UM0991
User manual

Developing and debugging your STM8L-DISCOVERY application code

Introduction

This document complements the information in the STM8L datasheets by describing the software environment and development recommendations required to build an application around the STM8L-DISCOVERY. It also explains how to use the STM8L firmware library provided by STMicroelectronics, in order to develop cost-effective applications.

In addition, ST provides a development package which can be used to build an application running on the STM8L-DISCOVERY. This package includes application code examples and a project template.

The STM8L-DISCOVERY is based on the STM8L15x MCU however it can be used to get to know the main features of other STM8L microcontrollers.

Reference documents

- STM8L-DISCOVERY user manual (UM0970)
- ST Visual Develop (STVD) user manual (UM0036)
- Discover STM8L15x power saving modes using the built-in I_{DD} measurement feature of the STM8L-Discovery (AN3269)
- Building a wave generator using the STM8L-DISCOVERY (AN3252)
- STM8L15x reference manual (RM0031)
- STM8L15xx datasheet

Contents

1 Overview of STM8 software development toolchains 6

1.1 ST Visual Develop (STVD) 6

1.2 ST Visual Programmer (STVP) 7

1.3 C and assembly compilers 7

1.4 IAR Embedded Workbench 8

1.5 Firmware libraries 8

2 Installing the development toolchain 9

2.1 Downloading and installing STVD 9

2.2 Downloading and installing the compilers 9

2.2.1 Installing the Cosmic compiler 9

2.2.2 Installing the Raisonance compiler 12

2.2.3 Compiler settings 15

2.3 Downloading and Installing IAR Embedded Workbench 16

3 Description of firmware library 21

3.1 STM8L15x standard firmware library 21

3.1.1 Description of the STM8L15x standard firmware library 22

3.1.2 STM8L15x standard firmware library online help 23

4 Configuring the option bytes for your application 24

5 Building and running your project 27

5.1 **Description of the STM8L-DISCOVERY development package 27**

5.1.1 Libraries 27

5.1.2 Application code examples 27

5.1.3 Project structure 28

5.2 Running the STM8L-Discovery examples 28

5.2.1 With STVD 28

5.2.2 With IAR Embedded Workbench 30

5.3 Creating your own project 31

5.3.1 Creating your project structure 31

5.3.2 Creating your STVD project 32

5.3.3 Linking the libraries to your STVD project 35
5.3.4 Creating your IAR Embedded Workbench project 38
5.3.5 Linking the libraries to your IAR project 41
5.4 Updating the libraries 44
5.4.1 Updating the STM8L15x standard firmware library 44

6 Building, debugging and running your application with STVD 45
6.1 Building your application 45
6.2 Debugging your application 46
6.3 Online help 48

7 Revision history 50

List of tables

Table 1. Document revision history 50

List of figures

Figure 1. STVD overview . . . 7
Figure 2. Cosmic compiler form . . . 10
Figure 3. Downloading Cosmic compiler software wizard . . . 11
Figure 4. Cosmic compiler installation wizard . . . 12
Figure 5. Cosmic compiler . . . 12
Figure 6. Raisonance compiler installation wizard . . . 13
Figure 7. Activation code registration . . . 14
Figure 8. Personal information form . . . 14
Figure 9. Serial key information form . . . 15
Figure 10. Selecting the Cosmic compiler for the Discover project . . . 16
Figure 11. Product Registration and Download form . . . 17
Figure 12. Download software . . . 18
Figure 13. Enter User Information . . . 19
Figure 14. Enter license key . . . 20
Figure 15. STM8L15x standard firmware library architecture . . . 22
Figure 16. stm8l15x_conf.h peripheral define statements . . . 23
Figure 17. STM8 firmware library online help home page . . . 23
Figure 18. Select your MCU . . . 24
Figure 19. STVP option byte selection menu . . . 25
Figure 20. STVP option byte programming menu . . . 26
Figure 21. STVP option byte programming message . . . 26
Figure 22. STM8L-Discovery_dev package content . . . 27
Figure 23. Selecting the debug instrument . . . 30
Figure 24. Selecting the debug instrument . . . 31
Figure 25. Creating your project structure . . . 32
Figure 26. Creating the STVD workspace for your project . . . 33
Figure 27. Checking the selected compiler . . . 33
Figure 28. Select the MCU . . . 34
Figure 29. Copy stm8l15x.h to Include Files/FWLib . . . 35
Figure 30. No library linked . . . 36
Figure 31. Linking the STM8L15x standard firmware library - step 1 . . . 36
Figure 32. Linking the STM815x standard firmware library - step 2 . . . 37
Figure 33. IAR Embedded Workbench workspace window . . . 39
Figure 34. Checking the Additional Include Directories path . . . 39
Figure 35. Select the MCU . . . 40
Figure 36. Copy stm8s.h to Include Files/FWLib . . . 41
Figure 37. No libraries linked . . . 42
Figure 38. Linking the STM8L standard firmware library - step 1 . . . 42
Figure 39. Linking the STM8L standard firmware library - step 2 . . . 43
Figure 40. Selecting the project configuration . . . 45
Figure 41. Building your project . . . 46
Figure 42. Building successful message . . . 46
Figure 43. Debugging your application . . . 47
Figure 44. Peripheral registers window . . . 48
Figure 45. Watch window . . . 48
Figure 46. STVD online help . . . 49

1 Overview of STM8 software development toolchains

To develop, compile and run an application software on an STM8L microcontroller, the following software toolchain components are required:

- Integrated development environment (IDE) composed of the ST Visual Develop (STVD) and the ST Visual Programmer software interface (STVP)
- Compilers
- Firmware libraries: they are optional, and allow to easily create a new application

STMicroelectronics provides a free software package (ST_toolset) including STVD and STVP.

You may also use the IAR Embedded Workbench for STMicroelectronics STM8. This toolchain is made up of the following components:

- Integrated development environment for building and debugging embedded applications.
- Compiler

Warning: IAR Embedded Workbench supports STM8L firmware library versions 1.3.0 and higher

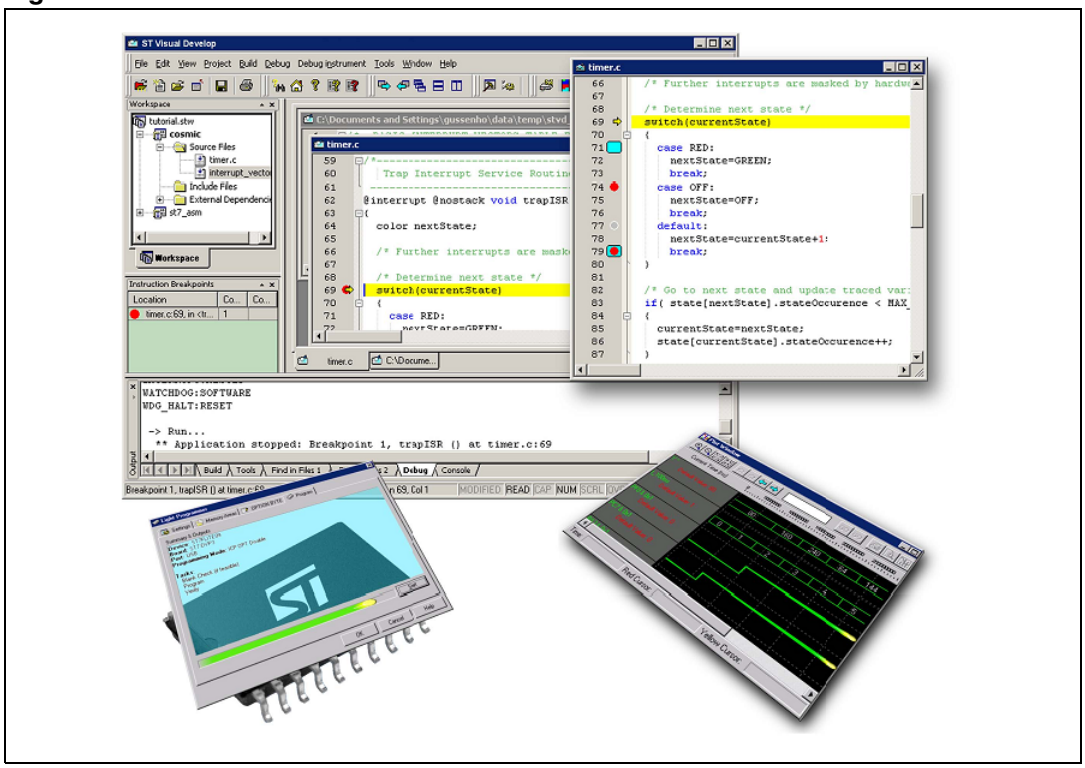
1.1 ST Visual Develop (STVD)

STVD is a full-featured development environment. It is a seamless integration of the Cosmic and Raisonance C compilers for STM8 microcontroller family. These compilers are free when developing code up to 16 Kbytes.

STVD main features are:

- Seamless integration of C and ASM compilers
- Full-featured debugger
- Project management
- Syntax highlighting editor
- Integrated programming interface

Figure 1. STVD overview



1.2 ST Visual Programmer (STVP)

STVP is a easy-to-use graphical interface allowing to read, write and verify the code and data programmed in your STM8 microcontroller Flash program memory, data EEPROM and option bytes. STVP also features a project mode for saving programming configurations and automating programming sequences.

1.3 C and assembly compilers

The C and assembly compilers are seamlessly integrated into the STVD development environment. They allow to directly configure and control the building of your application from an easy-to-use graphical interface.

The supported compilers are the following:

- Cosmic C compiler for STM8 (free version up to 16 Kbytes of code and temporary storage of up to 32 KBytes)

  For more information, refer to http://www.cosmic-software.com.
- Raisonance C compiler for STM8 (free up to 32Kbytes of code)

  For more information, refer to http://www.raisonance.com.
- STM8 assembler linker

  This is a free assembly toolchain included in the STVD toolset. It allows you to assemble and link your application source code.

1.4 IAR Embedded Workbench

IAR Embedded Workbench is a set of development tools for building and debugging embedded system applications using assembler and C. It provides a completely integrated development environment that includes a project manager, editor, build tools and C-SPY debugger. The C compiler for STM8 which is included in the toolset is free for up to 8Kbytes of code.

1.5 Firmware libraries

The STM8L15x standard firmware library is a complete package consisting of drivers for all the standard peripherals of STM8L15x microcontrollers. It is written in strict ANSI-C code and is fully MISRA C 2004 compliant.

Refer to *Section 3.1: STM8L15x standard firmware library* and for details on the installation and configuration of this library

2 Installing the development toolchain

2.1 Downloading and installing STVD

STVD software is available at http://www.st.com/stonline/products/support/micro/files/sttoolset.exe.

To install STVD, download the installation software and follow each step of the installation wizard.

When the installation is complete, the executable is available from ***START>Programs>ST Toolset>Developement Tools>ST Visual Develop*** or under **C:\Program Files\STMicroelectronics\st_toolset\stvd**.

2.2 Downloading and installing the compilers

Cosmic and Raisonance compilers are compatible with STVD. They are available together with their documentation at the following urls:

- Cosmic: http://www.cosmicsoftware.com/download_stm8_16k.php
- Cosmic: http://www.cosmicsoftware.com/download_stm8_32k.php
- Raisonance: http://www.mcu-raisonance.com

Please note that the Cosmic 32K version does not include direct support from Cosmic.

Note: A free license is required to use the compilers.

2.2.1 Installing the Cosmic compiler

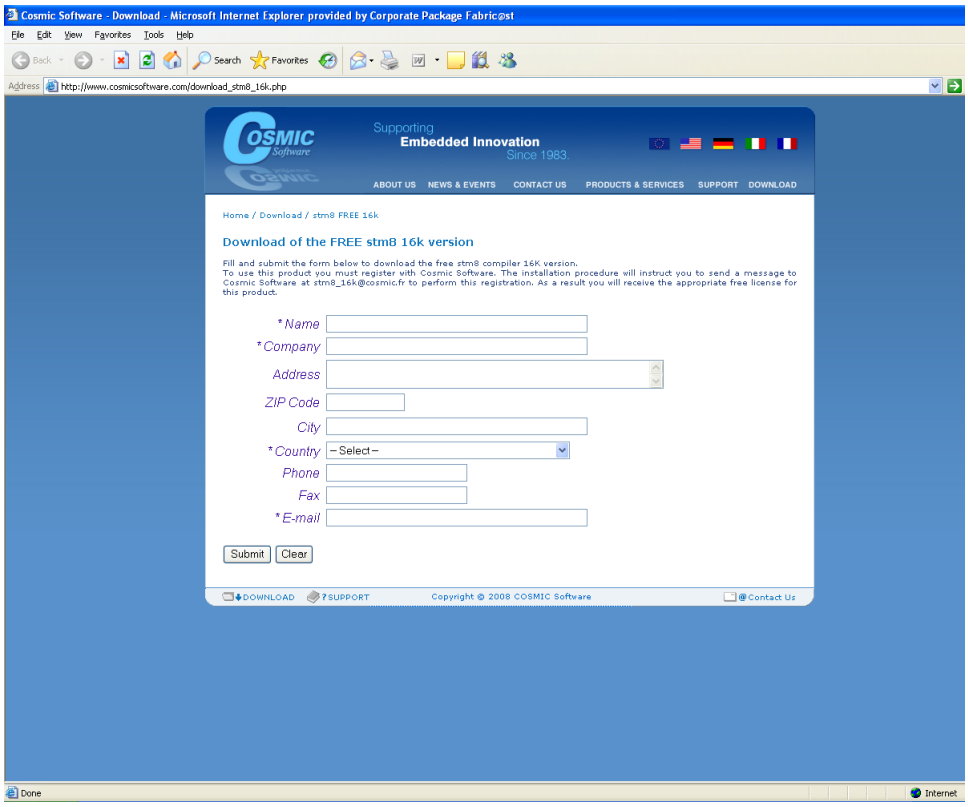
To install the Cosmic compiler, follow the sequence described below:

1. Step 1

   Connect to http://www.cosmicsoftware.com/download_stm8_16k.php, or to http://www.cosmicsoftware.com/download_stm8_32k.php and fill in the information form (see *Figure 2*).

   Click **Submit** to access the download page.

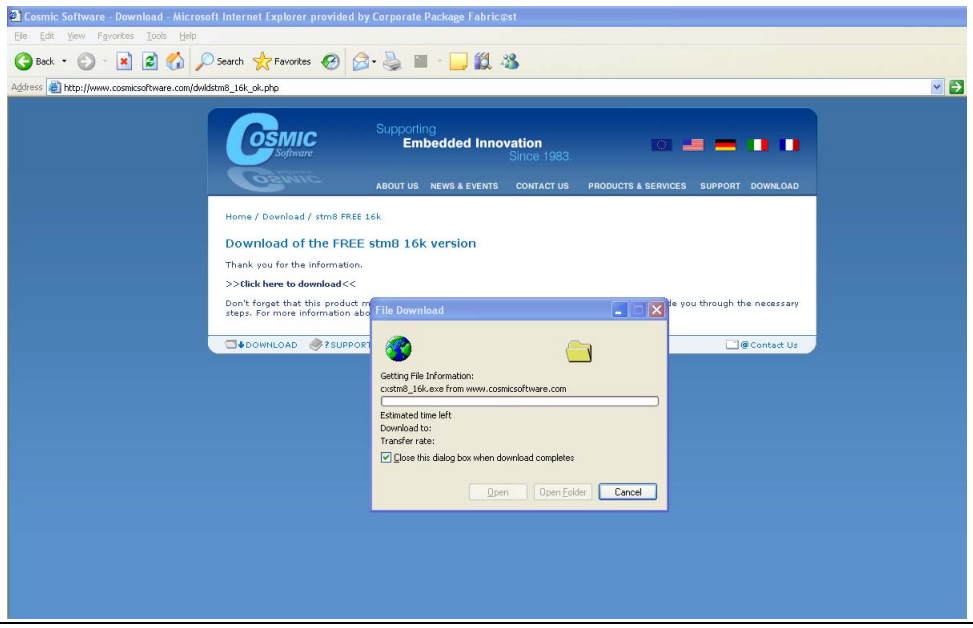
Figure 2. Cosmic compiler form


Cosmic Software - Download - Microsoft Internet Explorer provided by Corporate Package Fabric@st
File Edit View Favorites Tools Help
Address http://www.cosmicsoftware.com/download_stm8_16k.php
Supporting
Embedded Innovation
Since 1983.
ABOUT US NEWS & EVENTS CONTACT US PRODUCTS & SERVICES SUPPORT DOWNLOAD
Home / Download / stm8 FREE 16k
Download of the FREE stm8 16k version
Fill and submit the form below to download the free stm8 compiler 16K version.
To use this product you must register with Cosmic Software. The installation procedure will instruct you to send a message to Cosmic Software at stm8_16k@cosmic.fr to perform this registration. As a result you will receive the appropriate free license for this product.
*Name
*Company
Address
ZIP Code
City
*Country
– Select –
Phone
Fax
*E-mail
Submit
Clear
DOWNLOAD
SUPPORT
Copyright © 2008 COSMIC Software
Contact Us
Done
Internet

2. Step 2

   Download and run the installation software wizard (see *Figure 3*).

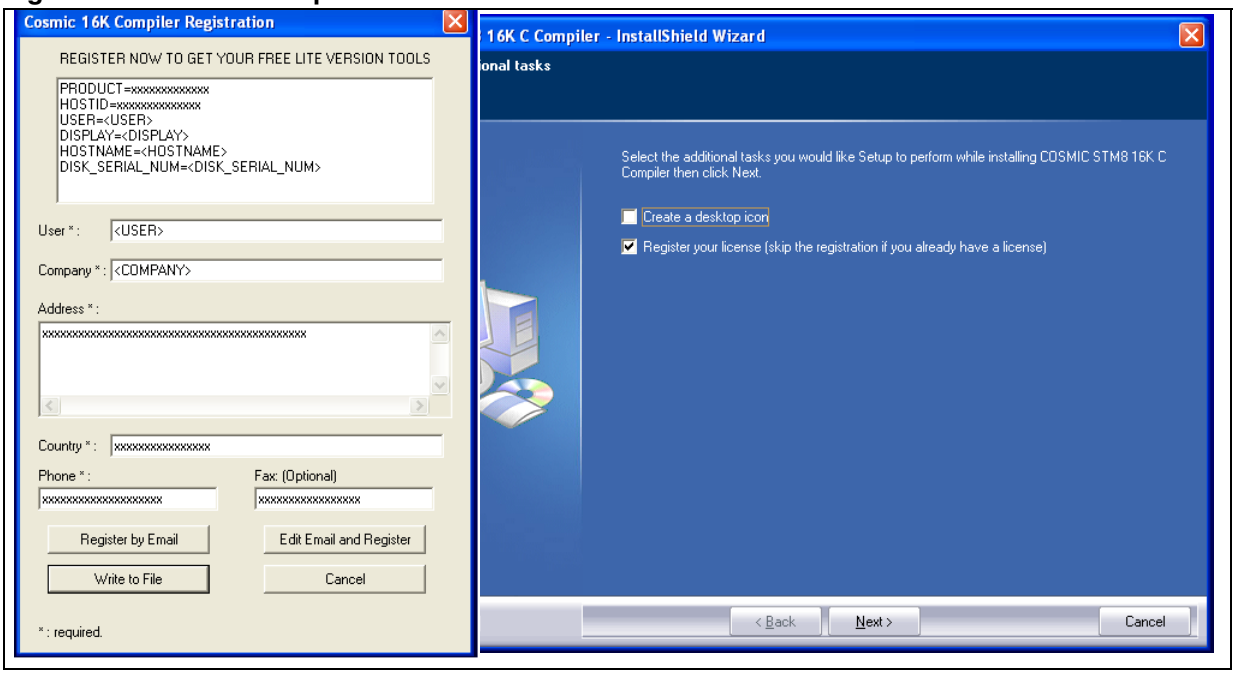
Figure 3. Downloading Cosmic compiler software wizard



3. Step 3

   Follow the wizard instructions to install the Compiler (see *Figure 4*). Do not forget to register for a free license and click **Register by Email**. You will receive the license file by e-mail. Note that the procedure is not done automatically, it may take a few hours or a few days (after the week-end), depending on your location and time zone, before you receive your license file.

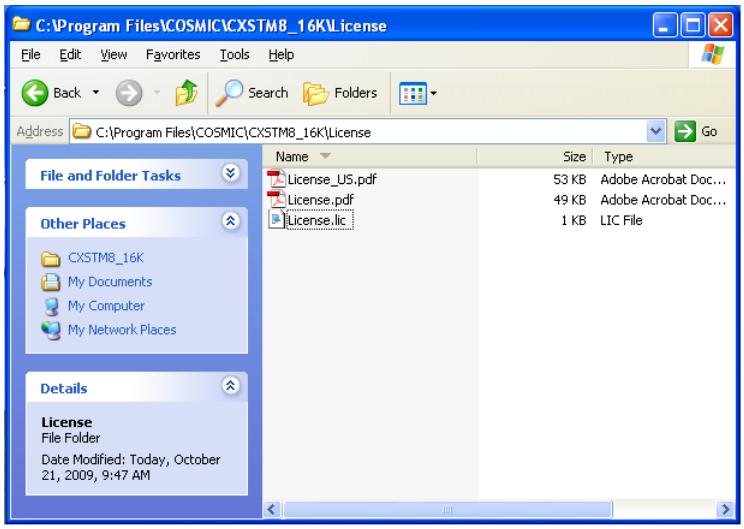
Figure 4. Cosmic compiler installation wizard



4. Step 4

   Copy the licence file that you have received into the corresponding directory ***\Program files\COSMIC\CXSTM8_16K\License*** *or* ***\Program files\COSMIC\CXSTM8_32K\License*** *(see Figure 5).*

Figure 5. Cosmic compiler



2.2.2 Installing the Raisonance compiler

To install the Raisonance compiler, follow the sequence described below:

1. Step 1

   Download Ride7 and the Raisonance kit RKit-STM8 from http://www.mcu-raisonance.com/microcontrollers-link-free-download.html, and follow the installation wizard instructions (see *Figure 6*).

Figure 6. Raisonance compiler installation wizard

Launcher
Ride7
The following components are currently installed on this machine:
Ride7 [7.20.09.0139]
Select the new components you want to install:
RKit-STM8 for Ride7
- ProductName: RKit-STM8 for Ride7
- Version: 2.24.09.0238
Use the following web address to get technical support and updates:
http://www.raisonance.com
Select the destination directory for Ride7 IDE and its components:
C:\Program Files\Raisonance\Ride
...
Available Space: 12987 MB
Start install
Close

2. Step 2 - Register for a free 32 Kbyte license

   a) You need a computer serial number. To get it, open Ride, click **Help**>**Licence**, and enter your name, you company, and select **Manual Activation**. The computer serial key will appear as shown in *Figure 7*.

   b) To get your free 32Kbyte activation code, go to http://www.mcu-raisonance.com/stm8_registration.html (see *Figure 7*). Fill the question form with your personal information (see *Figure 8*), and the registration form with your serial key (see *Figure 9*).

   After this operation, you will receive your activation code by e-mail

   Copy the activation code received by e-mail into installation window (see *Figure 7*) and you will be granted a free 32Kbyte Raisonance compiler licence to be used by STVD.

Figure 7. Activation code registration

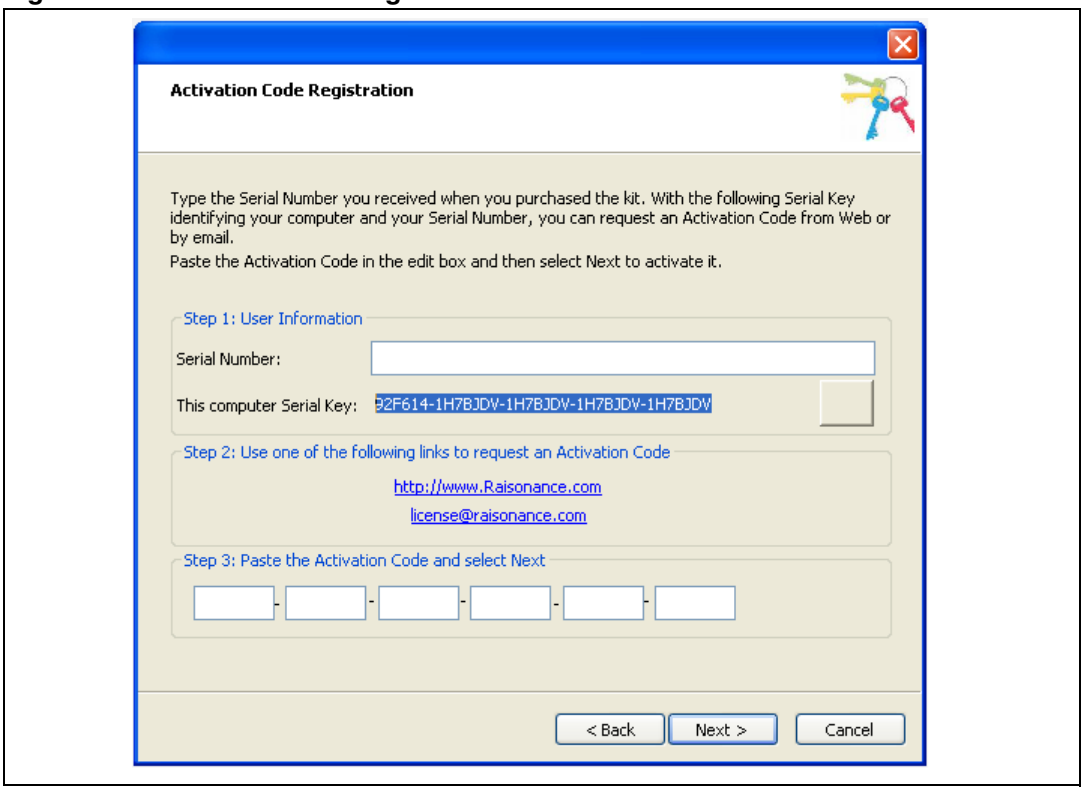

Activation Code Registration
Type the Serial Number you received when you purchased the kit. With the following Serial Key identifying your computer and your Serial Number, you can request an Activation Code from Web or by email.
Paste the Activation Code in the edit box and then select Next to activate it.
Step 1: User Information
Serial Number:
This computer Serial Key:
92F614-1H7BJDV-1H7BJDV-1H7BJDV-1H7BJDV
Step 2: Use one of the following links to request an Activation Code
http://www.Raisonance.com
license@raisonance.com
Step 3: Paste the Activation Code and select Next
< Back
Next >
Cancel

Figure 8. Personal information form

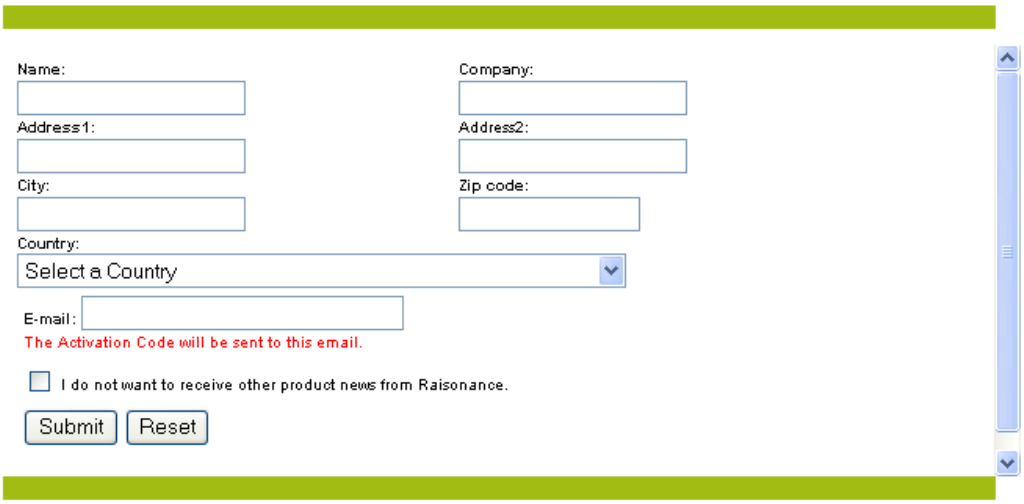
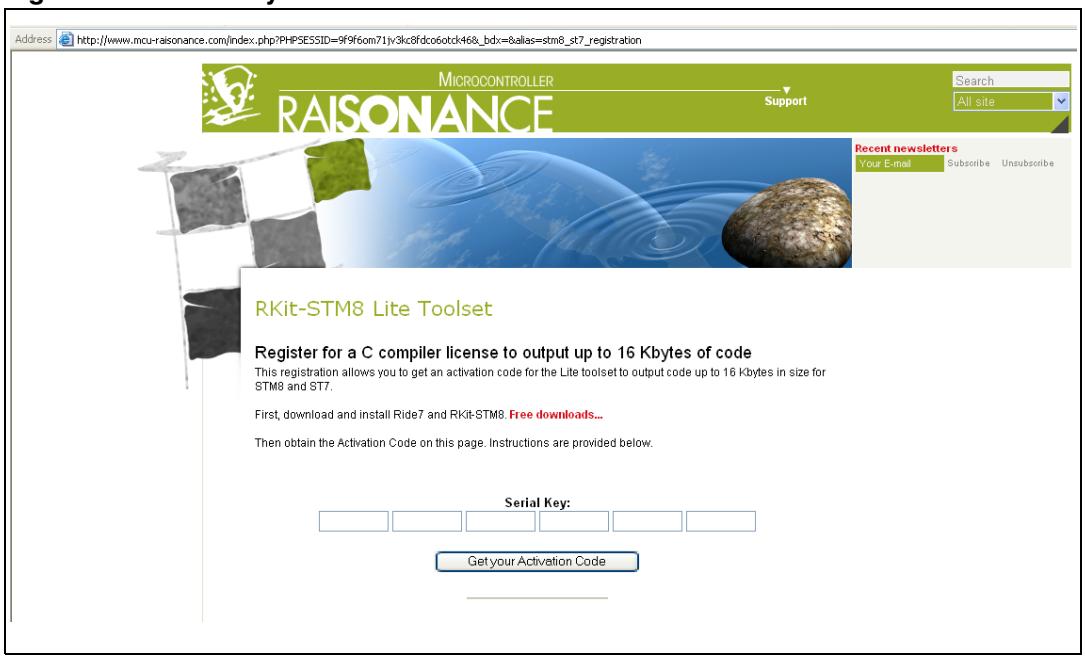


Figure 9. Serial key information form


Address
http://www.mcu-raisonance.com/index.php?PHPSESSID=9f9f6om71jv3kc8fdco6otck46&_bdx=&alias=stm8_st7_registration
MICROCONTROLLER
RAISONANCE
Support
Search
All site
Recent newsletters
Your E-mail
Subscribe
Unsubscribe
RKit-STM8 Lite Toolset
Register for a C compiler license to output up to 16 Kbytes of code
This registration allows you to get an activation code for the Lite toolset to output code up to 16 Kbytes in size for STM8 and ST7.
First, download and install Ride7 and RKit-STM8. Free downloads...
Then obtain the Activation Code on this page. Instructions are provided below.
Serial Key:
Get your Activation Code

2.2.3 Compiler settings

STVD uses the default compiler defined at the first launch of the toolchain. This compiler is not defined during the installation.

You can select a different compiler for a specific project in the **General tab** of the STVD **Project**>**Settings** window. Browse to the root path depending on the the compiler you want to use (see *Figure 10*) :

Root pathes for compilers are :

- Cosmic 16K

C:\Program Files\COSMIC\CXSTM8_16K

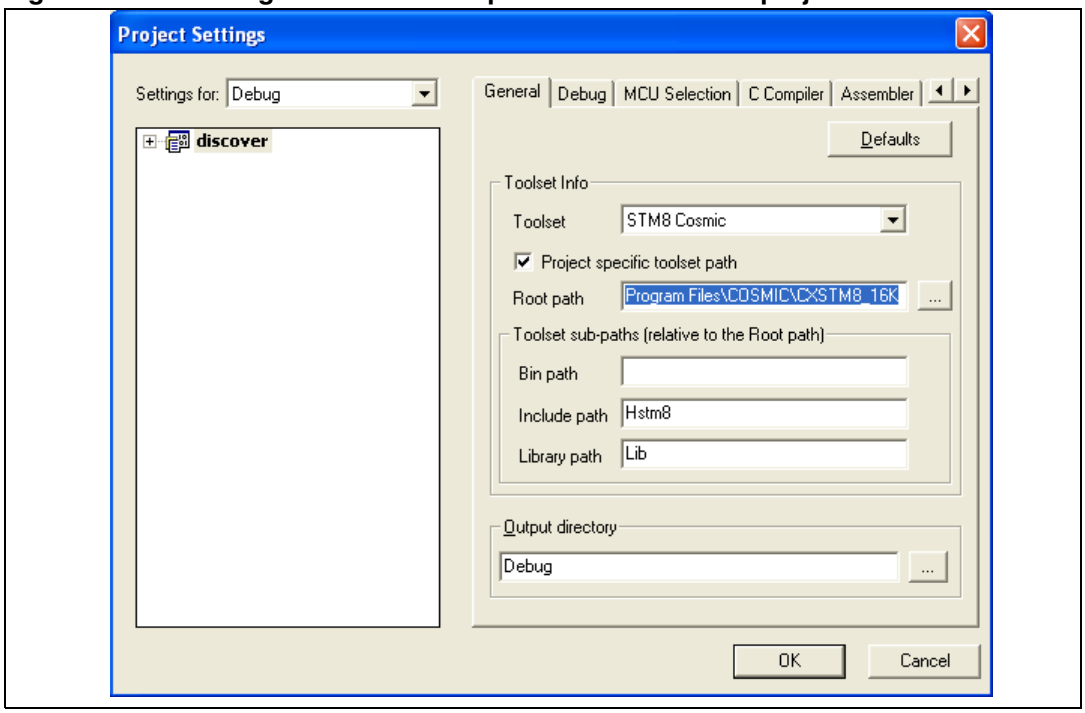
- Cosmic 32K

C:\Program Files\COSMIC\CXSTM8_32K

- Raisonance 32K

C:\Program Files\Raisonance\Ride

Figure 10. Selecting the Cosmic compiler for the Discover project



2.3 Downloading and Installing IAR Embedded Workbench

IAR Embedded Workbench for STM8 is available at http://www.iar.com in the **Downloads** tab. It integrates the IAR C Compiler and assembler.

To install IAR Embedded Workbench, follow the sequence described below:

1. Step1

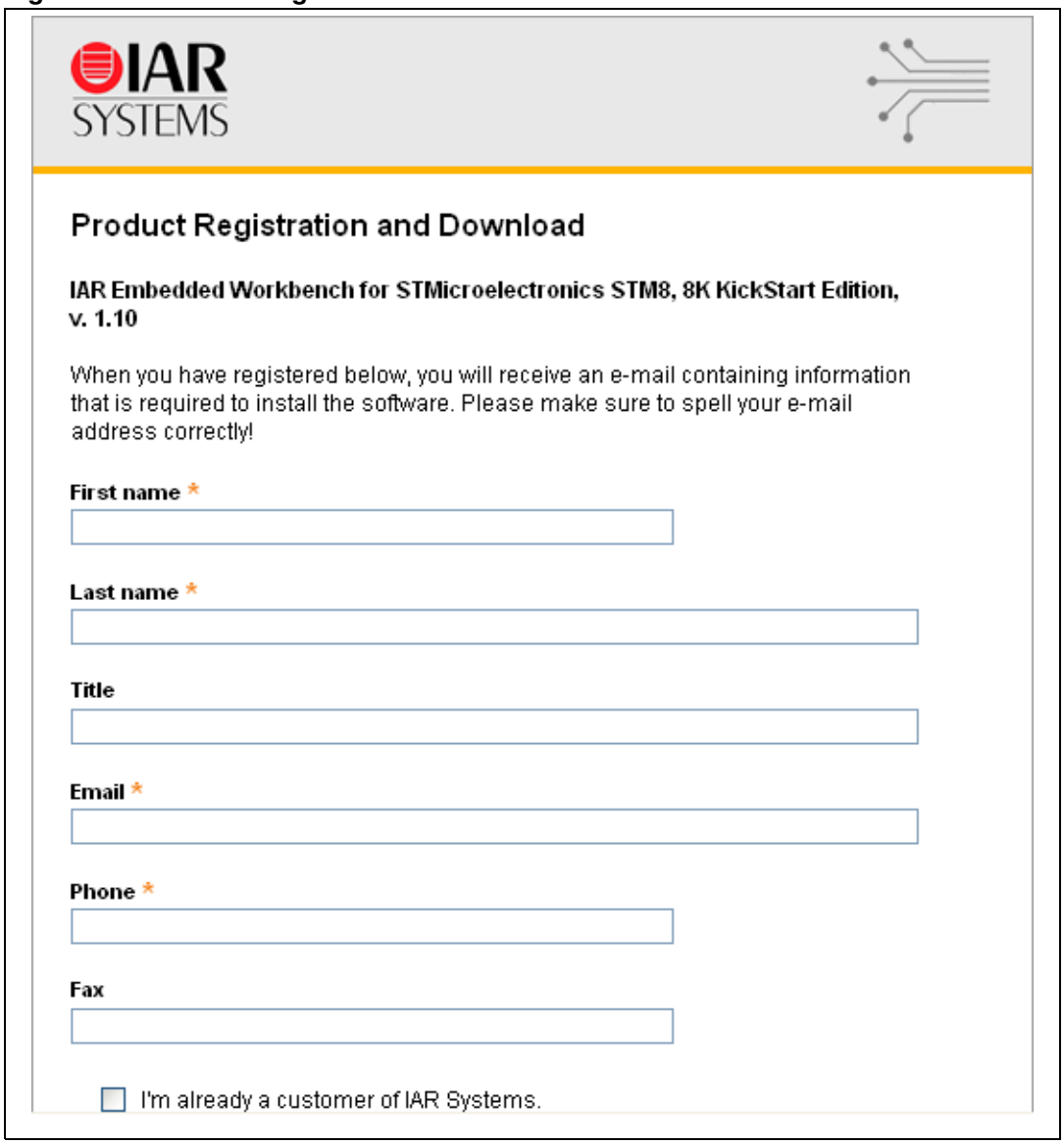
   Download the installation software, either the 30-day evaluation edition or the 8 KB-limited Kickstart edition according to your requirements, from http://www.iar.com and follow the installation wizard instructions.

Note: The procedure described here is for the 8 KB-limited Kickstart edition installation only.

2. Step1

   Fill in the **Product Registration and Download** form (*Figure 11*) and click **Submit Registration**. An email asking you to confirm the registration is then sent to the address you have specified in the registration form. It might take some time before to receive the email, as the operation is manual.

Figure 11. Product Registration and Download form


IAR SYSTEMS
Product Registration and Download
IAR Embedded Workbench for STMicroelectronics STM8, 8K KickStart Edition, v. 1.10
When you have registered below, you will receive an e-mail containing information that is required to install the software. Please make sure to spell your e-mail address correctly!
First name *
Last name *
Title
Email *
Phone *
Fax
I'm already a customer of IAR Systems.

3. Step 3

   Open the mail and click the URL. A web page opens with the License Number and the License Key. This page also give you information on how to download and install the product. (see *Figure 12*)

4. Step 4

   To download the product, click the **Europe (HTTP site)** URL. (see *Figure 12*)

   Once IAR Embedded Workbench downloaded, the installation process starts automatically.

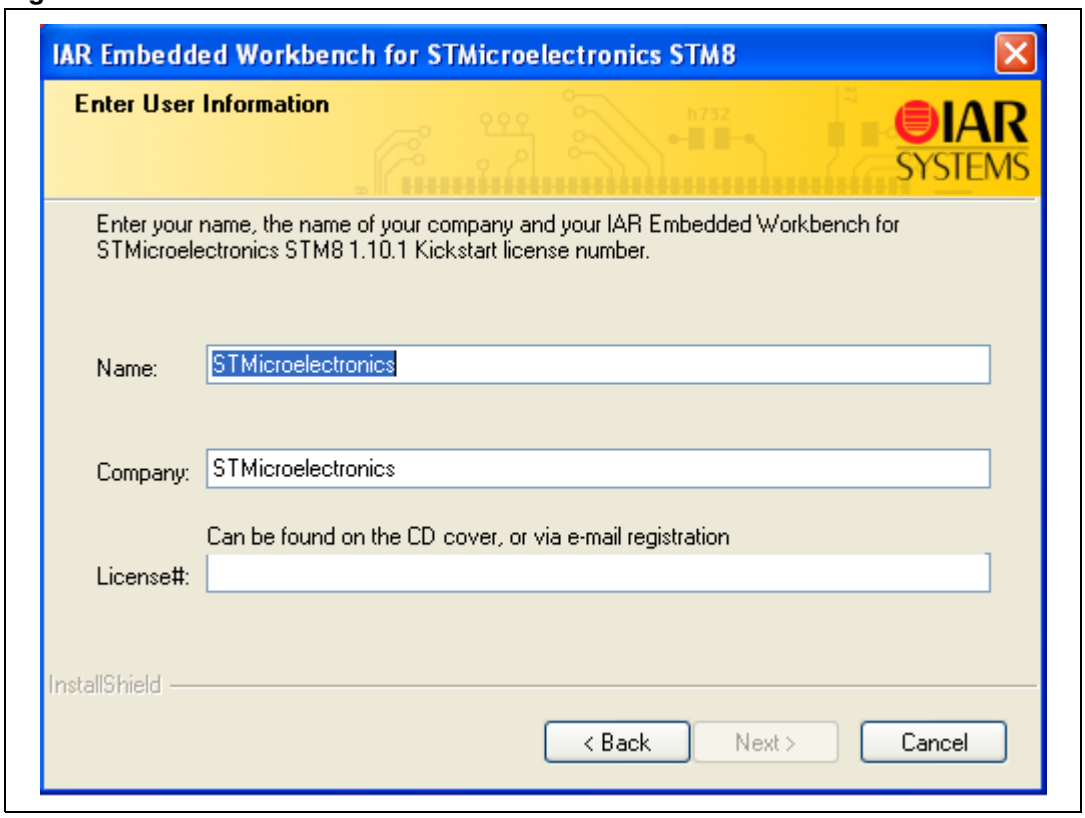
Figure 12. Download software

IAR SYSTEMS
Registration Confirmed
Thank you for your registration of the product:
IAR Embedded Workbench for STMicroelectronics STM8, 8K KickStart Edition, v. 1.10
Download software (EXE, 78 MB)
Download from:
Europe (HTTP site)
Europe (FTP site)
File Download - Security Warning
Do you want to run or save this file?
Name: EWSTM8-KS-1101.exe
Type: Application, 78,0 MB
From: supp.iar.com
Run
Save
Cancel
While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?
If you prefer, you can have the License Number and Key sent to you by email for later reference.

5. Step 5

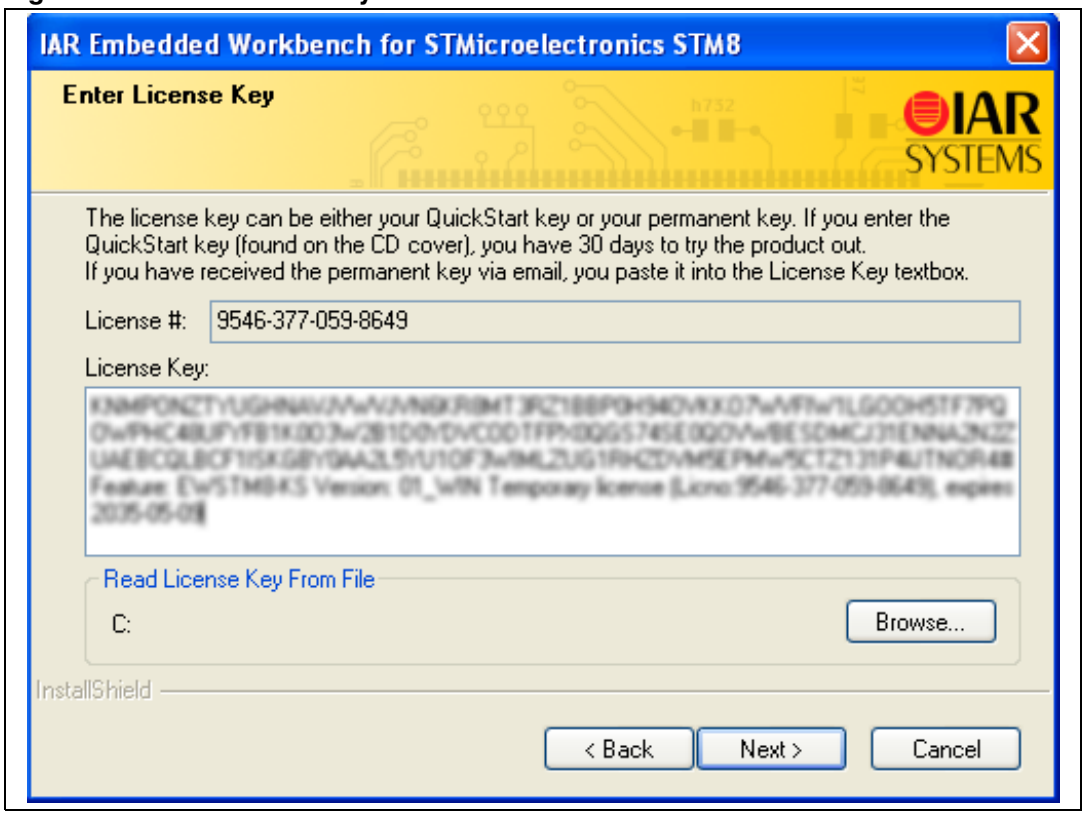
   Go through the installation steps and copy the License Number to the **License#** field (see *Figure 13*)

Figure 13. Enter User Information


IAR Embedded Workbench for STMicroelectronics STM8
Enter User Information
IAR SYSTEMS
Enter your name, the name of your company and your IAR Embedded Workbench for STMicroelectronics STM8 1.10.1 Kickstart license number.
Name:
STMicroelectronics
Company:
STMicroelectronics
Can be found on the CD cover, or via e-mail registration
License#:
InstallShield
< Back
Next >
Cancel

6. Step 6
   Enter your License Key in the License Key textbox (see *Figure 14*)
   Click **Next** to finish the installation process.

Figure 14. Enter license key


IAR Embedded Workbench for STMicroelectronics STM8
Enter License Key
IAR SYSTEMS
The license key can be either your QuickStart key or your permanent key. If you enter the QuickStart key (found on the CD cover), you have 30 days to try the product out.
If you have received the permanent key via email, you paste it into the License Key textbox.
License #:
9546-377-059-8649
License Key:
Read License Key From File
C:
Browse...
InstallShield
< Back
Next >
Cancel

3 Description of firmware library

The STM8L15x microcontroller family is provided with the STM8L15x standard firmware library. It is strongly recommended to use the library to develop your project as proposed in the the STM8L-Discovery development package and implemented in the Discover project (see *Section 5.1: Description of the STM8L-DISCOVERY development package*).

The development package also contains the WavesGenerator project which don't use the STM8L standard firmware library except for the LCD driver.

The STM8L standard firmware library is part of the STM8L-DISCOVERY development package that is available from the STM8L-DISCOVERY web page at http://www.st.com/stm8l-discovery.

Warning: Make sure you have the latest version of the STM8L standard firmware library (see http://www.st.com/mcu).

3.1 STM8L15x standard firmware library

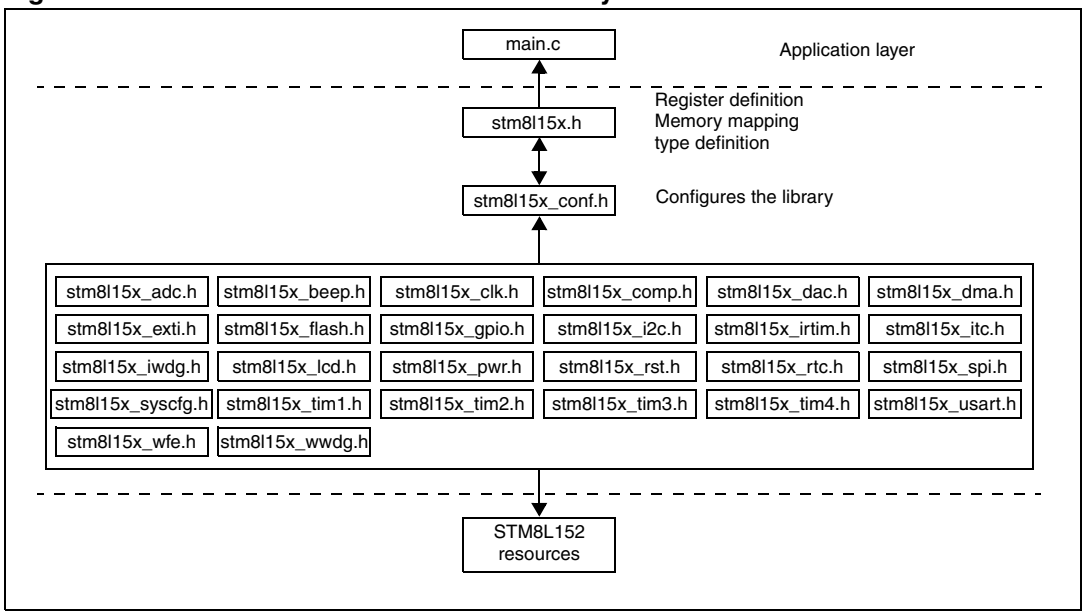
The STM8L15x standard firmware library contains a collection of routines, data structures, and macros covering the features of the STM8L15x peripherals, as well as a description of the device drivers (see *Figure 15*).

The STM8L15x standard firmware library allows to develop an application on any STM8L15x device without the need for in-depth study of each peripheral specifications. It saves significant time that would otherwise be spent in coding, while simultaneously reducing application development and integration cost.

It contains a set of examples for each peripheral. All these examples are provided with workspace and project definition files for STVD and C compilers and for IAR Embedded Workbench for STM8 to allow loading and compiling them easily into you development environment. These examples are developed to run on an STM8L15x device with STMicroelectronics STM8L1526-EVAL evaluation board.

They have to be considered as example codes to develop your own application. Some examples have already been tailored to run on the STM8L-DISCOVERY. They are available from the STM8L-DISCOVERY web pages at http://www.st.com/stm8l-discovery.

Figure 15. STM8L15x standard firmware library architecture



3.1.1 Description of the STM8L15x standard firmware library

The *stm8l15x.h* header file contains the definitions of constants and register structures for all peripherals. Uncomment `#define USE_STDPERIPH_DRIVER` when using the STM8L15x standard firmware library. It also includes common types definition used by the peripheral drivers. In addition, stm8l15x.h must be included in your main() routine.

The *stm8l15x_conf.h* file of the STM8L15x standard firmware library is used to configure the library by enabling the peripheral functions that are only used by your application (see *Figure 16*).

The peripheral interrupt function file, stm8l15x_it.c, must be modified to include the code to handle the interrupts used by your application.

Each peripheral driver is made up of the following files:

- The source code *stm8l15x_<periph>.c* containing all the software functions required to use the corresponding peripheral.
- The header *stm8l15x_<periph>.h* including the peripheral function prototypes as well as the variables, constant and structures used within these functions.

The flow that must be followed to create your application software using the STM8L15x standard firmware library is described in *Section 5.3: Creating your own project*.

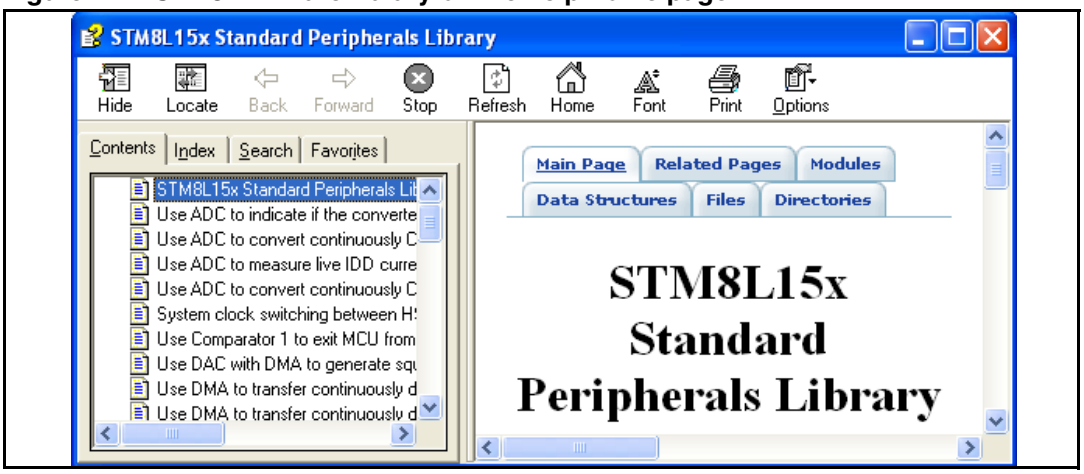
Figure 16. *stm8l15x_conf.h* peripheral define statements

```
/* Uncomment the line below to enable peripheral header file inclusion */
/* #include "stm8l15x_adc.h" */
/* #include "stm8l15x_beep.h" */
/* #include "stm8l15x_clk.h" */
/* #include "stm8l15x_comp.h" */
/* #include "stm8l15x_dac.h" */
/* #include "stm8l15x_dma.h" */
/* #include "stm8l15x_exti.h" */
/* #include "stm8l15x_flash.h" */
#include "stm8l15x_gpio.h"
/* #include "stm8l15x_i2c.h" */
/* #include "stm8l15x_irtim.h" */
/* #include "stm8l15x_itc.h" */
/* #include "stm8l15x_iwdg.h" */
/* #include "stm8l15x_lcd.h" */
/* #include "stm8l15x_pwr.h" */
/* #include "stm8l15x_rst.h" */
/* #include "stm8l15x_rtc.h" */
/* #include "stm8l15x_spi.h"*/
/* #include "stm8l15x_syscfg.h" */
/* #include "stm8l15x_tim1.h" */
/* #include "stm8l15x_tim2.h" */
/* #include "stm8l15x_tim3.h" */
/* #include "stm8l15x_tim4.h" */
/* #include "stm8l15x_usart.h" */
/* #include "stm8l15x_wfe.h" */
```

3.1.2 STM8L15x standard firmware library online help

An online help, *STM8L15x_stdperiph_lib_um.chm,* is available inside the firmware installation directory to help you with the structure of the library (see *Figure 17*).

Figure 17. STM8 firmware library online help home page



4 Configuring the option bytes for your application

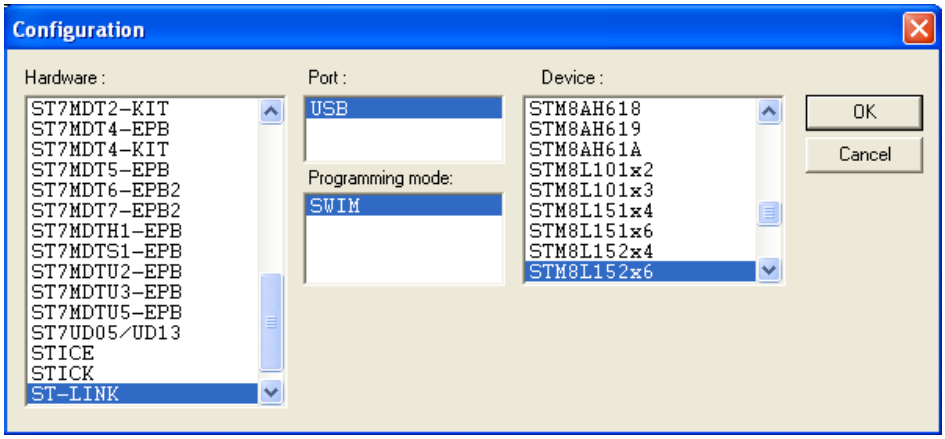
The option bytes allow to configure the device hardware features and memory protection. They are stored in a dedicated memory block. Refer to the option bytes section of the STM8L152x6 datasheet for a detailed description.

STVP can be used to program the STM8L option bytes. It is part of the free software package provided by STMicroelectronics. This easy-to-use tool has a graphical interface allowing to read, write and verify the STM8 Flash programming memory, data EEPROM and option bytes.

The STVP executable is available from the Windows **start** menu.

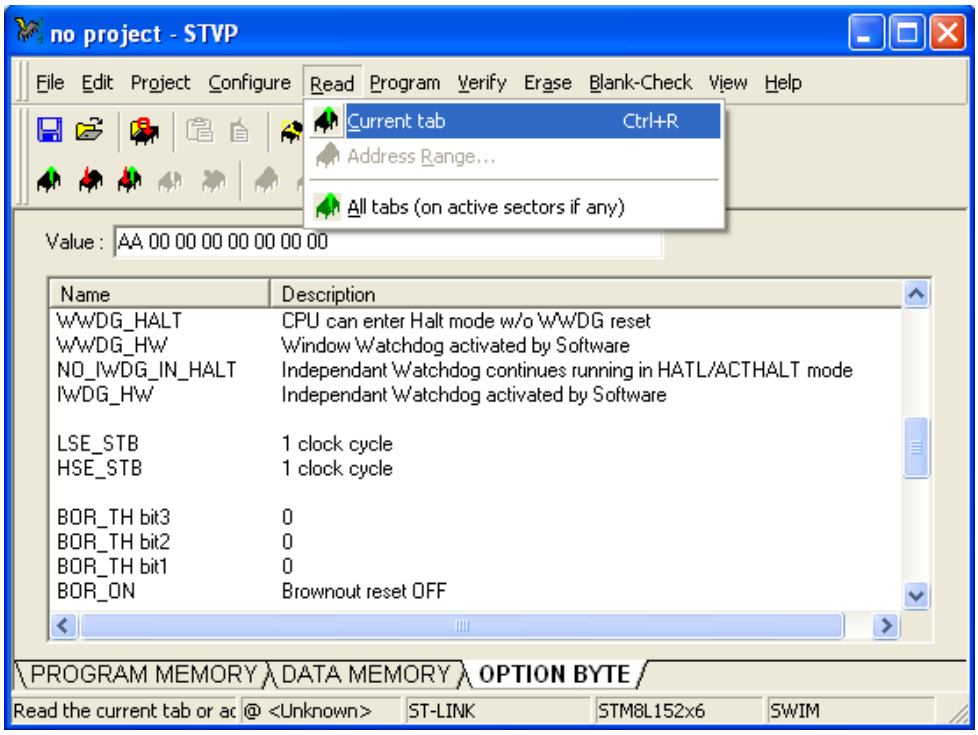
1. Open the STVP GUI and select **Configure>Configure ST Visual Programmer** from the main menu toolbar. The configuration dialog box opens.
2. In the **hardware** list, select ST-LINK as programming board.
3. In the **port** list, select USB as the host PC port to which the ST-LINK is connected. Only the ports that are compatible with the selected hardware are displayed.
4. in the **programming mode** list, select SWIM. The programming modes displayed depend on your programming equipment.
5. In the **device** list, select STM8L152x6 as the ST microcontroller you are going to program (see *Figure 18*).

Figure 18. Select your MCU



6. Select the **OPTION BYTE** tab, and click **Read >Current tab** in the menu toolbar. All the STM8L152C6T6 option bytes and their respective values are displayed in the window (see *Figure 19*).

Figure 19. STVP option byte selection menu



7. To program an option byte to a new value:
   a) Click the option byte description, and select the value.
   b) Click **Program>Current tab** in the main menu toolbar (see *Figure 20*).
   c) If the operation has completed successfully, a confirmation message is displayed in the Output window (see *Figure 21*).

Figure 20. STVP option byte programming menu

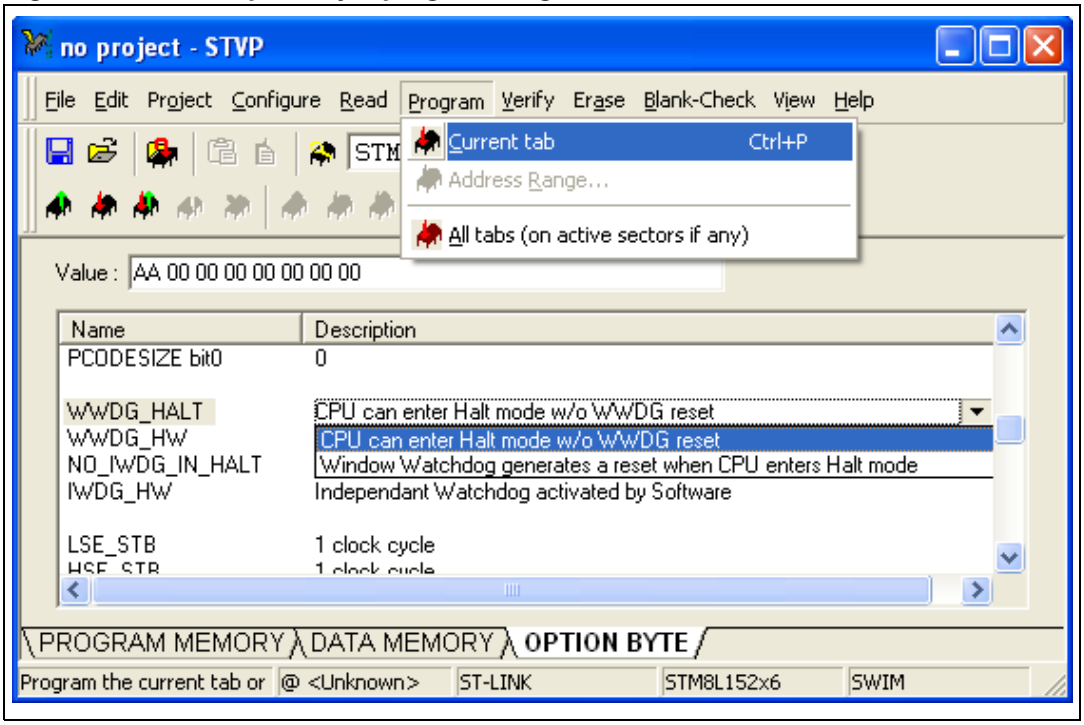
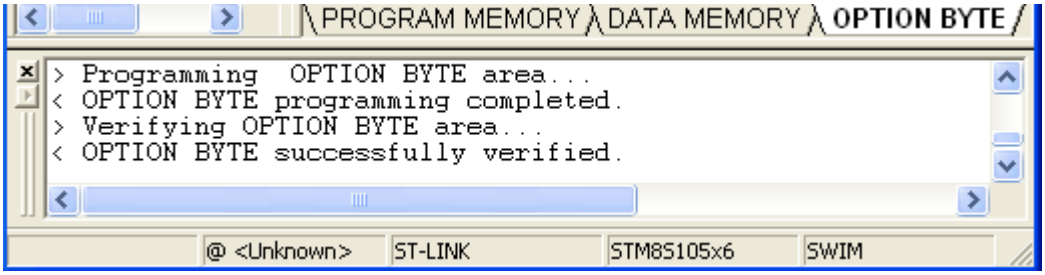


Figure 21. STVP option byte programming message



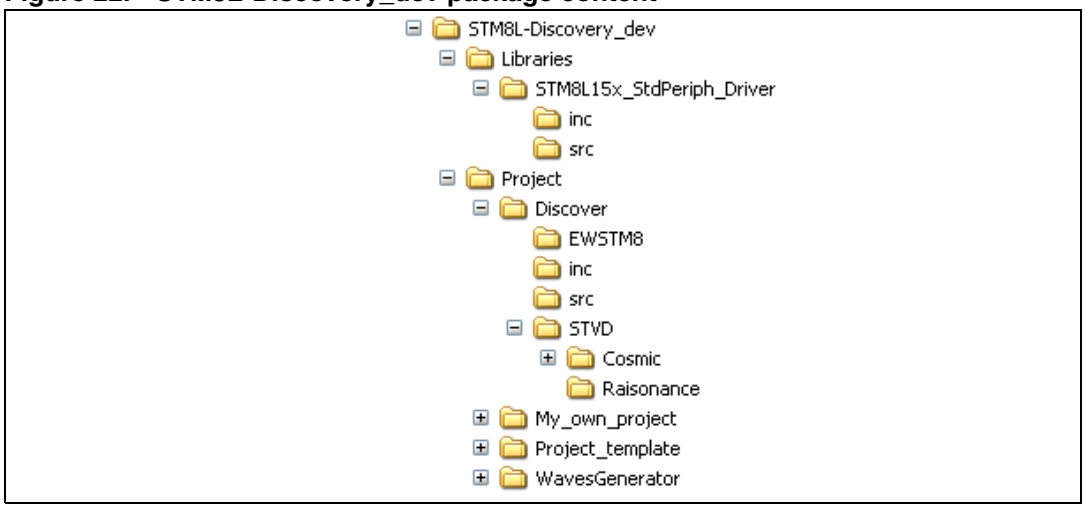
5 Building and running your project

5.1 Description of the STM8L-DISCOVERY development package

ST provides a complete development package, the STM8L-Discovery_dev. It allows to easily get started with the STM8L-DISCOVERY by simplifying application code development and execution using the standard peripherals drivers. This package can be downloaded together with the present user manual from http://www.st.com/STM8L-discovery.

The package is structured as follow:

Figure 22. STM8L-Discovery_dev package content



5.1.1 Libraries

The library is located in the **Libraries** directory.

- STM8L15x_StdPeriph_Driver

  This directory contains the source and header files of the STM8L standard firmware library

---

Warning: **It is strongly recommended to upgrade the library as soon as new releases are available (see *Section 5.4*).**

5.1.2 Application code examples

The application code examples are located in the **Project** directory:

- Discover

  The Discover demonstration firmware comes preprogrammed in the STM8L-DISCOVERY. This demo displays the STM8L152C6T6 Idd consumption on the STM8L-Discovery LCD while the MCU is in Run or in low power saving modes. This example uses the STM8L15x standard firmware library. For more details on the Discover application code implementation and library configuration, refer to application note

AN3269 *Discover STM8L15xx power saving modes using the built-in Idd measurement feature of STM8L-Discovery*. The Discover application project is compatible both with Cosmic, Raisonance compilers and also with IAR Embedded Workbench.

- WavesGenerator

  This application transforms the STM8L-Discovery into a basic wave generator. This code has been optimized and use only the LCD driver of the library. Refer to application note AN3252 *Building a wave generator using the STM8L-DISCOVERY* for details on the WavesGenerator application software code implementation and library configuration.

  The WavesGenerator project is compatible both with Cosmic and Raisonance compilers and also with IAR Embedded Workbench.

- Project_template

  This directory contains a standard project template that should be used to start any application software development.

5.1.3 Project structure

Each project is structured as follows:

- **inc**

  This directory contains all the header files for your application, including the configuration files for the standard firmware library (*STM8L15x_conf.h*). These files are used to tailor the library for your application.

- **src**

  This directory contains all the source files dedicated to your application code, such as *main.c*, stm8l15x_it.c, and *stm8_interrupt_vector.c* (interrupt mapping file used only for Cosmic compiler).

- **STVD\Cosmic (or \Raisonance)**

  This directory is used to store STVD workspace files (for example *discover.stw*). It includes the project and workspace for the Cosmic or Raisonance compiler, depending on the compiler you have selected.

- **EWSTM8**

  This directory is used to store IAR Embedded Workbench workspace files (for example discover.eww). It also include the project file (for example discover.ewp).

5.2 Running the STM8L-Discovery examples

Before starting developing your own application, run one of the examples provided with the development package to check if is is correctly configured. Take care to read the example-related documentation before proceeding because some examples may require additional hardware to work properly.

5.2.1 With STVD

Follow the steps below:

1. Install the development toolchain (see *Section 2: Installing the development toolchain*)
   – Cosmic or Raisonance compiler

– STVD

2. Download the STM8L-Discovery_dev.zip file and extract the directory **STM8L-Discovery_dev** on your PC (see *Section 5.1: Description of the STM8L-DISCOVERY development package*)
3. Run STVD

   From **START>Programs>ST Toolset>Development Tools>ST Visual Develop**

   a) Select **File -> Open Workspace**

   b) Browse for the project file .stw of the example you intend to run according to the compiler you have installed:

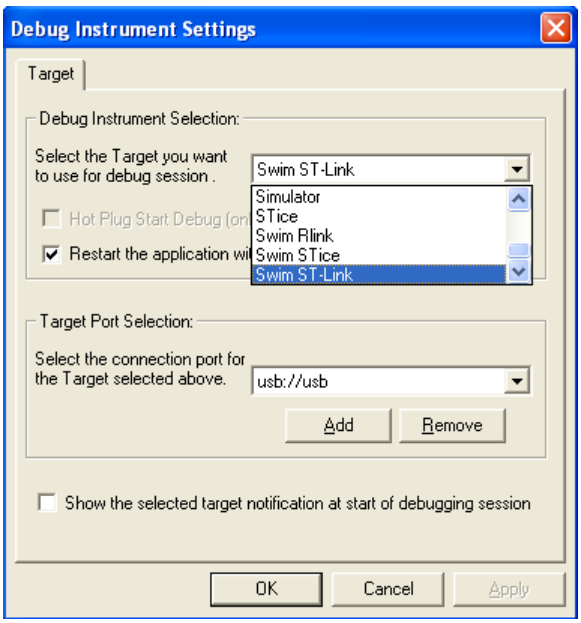
   – Discover example: *STM8L-Discovery_dev\Project\Discover\STVD\Cosmic\Discover.stw or STM8L-Discovery_dev\Project\Discover\STVD\Raisonance\Discover.stw*

   – WavesGenerator example: *STM8L-Discovery_dev\Project\WavesGenerator\STVD\Cosmic\WavesGenerator.stw* or *STM8L-Discovery_dev\Project\WavesGenerator\STVD\Raisonance\WavesGenerator.stw*
4. Configures STVD to use ST-LINK as debug instrument (see *Figure 23*):

   a) Click **Debug instrument -> Target Settings**.

   b) Select target **Swim ST-LINK** in the target list.
5. Click **Build -> Rebuild All** to build your application.
6. Clicking **Debug -> Start Debugging** to start the debug session.
7. Click **Debug -> Run** to run you application.
8. The application code example you have selected is now running. Check if it operates in compliance with what is described in the associated application note. You can also use STVD debug capabilities to run the code step by step and monitor the variables (see *Section 6.2: Debugging your application*).

Figure 23. Selecting the debug instrument



5.2.2 With IAR Embedded Workbench

Follow the steps below:

1. Install the development toolchain (see *Section 2: Installing the development toolchain*)
   – IAR Embedded Workbench for STM8
2. Download the STM8L-Discovery_dev.zip file and extract the directory **STM8L-Discovery_dev** on your PC (see *Section 5.1: Description of the STM8L-DISCOVERY development package*)
3. Run IAR Embedded Workbench

   From **START>Programs>IAR Systems>IAR Embedded Workbench for STMicroelectronics STM8>IAR Embedded Workbench**

   a) Select **File -> Open -> Workspace**

   b) Browse for the project file .eww of the example you intend to run :

   – Discover example: *STM8L-Discovery_dev\Project\Discover\EWSTM8\Discover.eww*

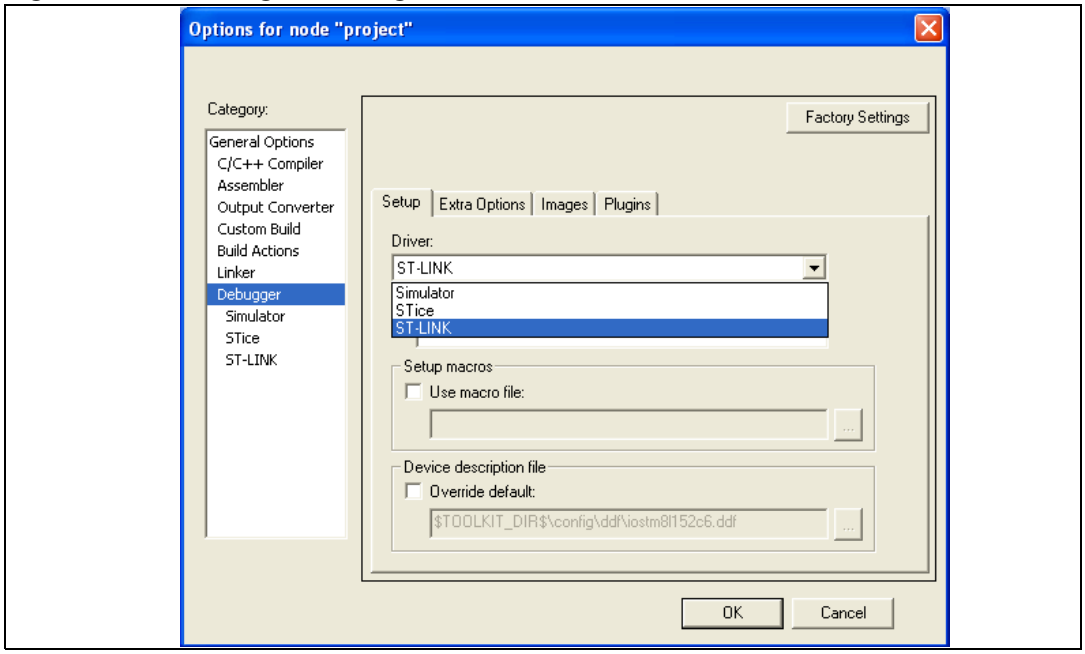
   – WavesGenerator example: *STM8L-Discovery_dev\Project\WavesGenerator\EWSTM8\WavesGenerator.eww*
4. Configures IAR to use ST-LINK as debug instrument (see *Figure 23*):

   a) Right click your project name ??? then select Options.

   b) Select Driver **ST-LINK** in the setup tab of the Debugger Category
5. Click **Project -> Rebuild All** to build your application.
6. Clicking **Project -> Download and Debug** to start the debug session.
7. Click **Debug -> Go** to run you application.
8. The application code example you have selected is now running. Check if it operates in compliance with what is described in the associated application note. You can also use

IAR debug capabilities to run the code step by step and monitor the variables (see IAR Embedded Workbench help).

Figure 24. Selecting the debug instrument



5.3 Creating your own project

This section explains step by step how to create your own application project using either STVD or IAR Embedded Workbench environments. The firmware library can be used or not according to the kind of application code to develop.

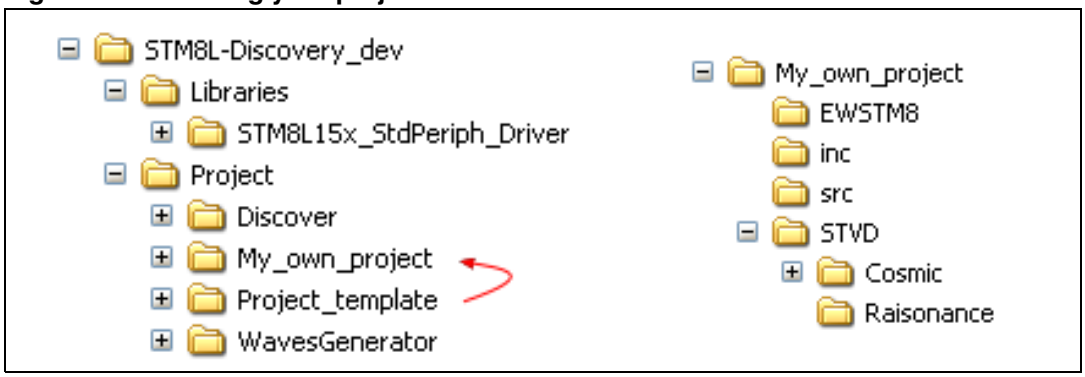
All projects must be created starting from STM8L-Discovery_dev development package (see *Section 5.1: Description of the STM8L-DISCOVERY development package*).

5.3.1 Creating your project structure

The best way to proceed is to start from the **Project_template** directory:

1. Extract the content of STM8L-Discovery_dev.zip file on your PC.
2. Copy the **Project_template** directory and rename it **My_own_project** (see *Figure 25*). Your STVD or IAR project comes already structured thus simplifying the project creation.

Figure 25. Creating your project structure

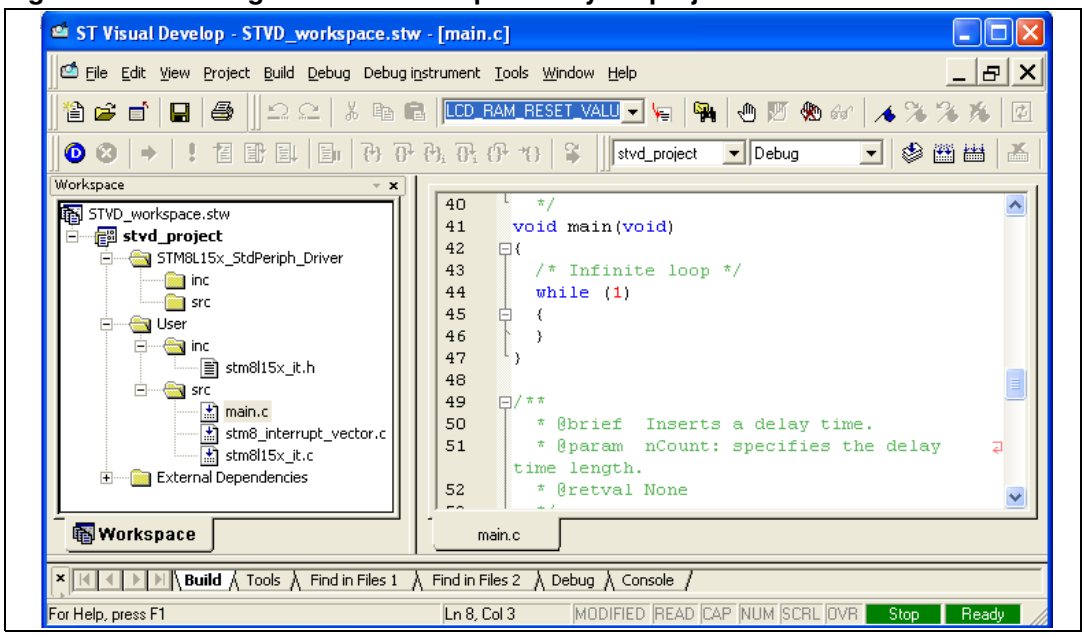


5.3.2 Creating your STVD project

1. Open STVD and click **File>Open Workspace**
2. Browse for the **Workspace location** and select **STVD_workspace.stw** in the **Cosmic** or **Raisonance** directory depending on the compiler you intend to use:
   - For Cosmic, select **\STM8L-DISCOVERY_dev\Project\My_own_project\STVD\Cosmic\STVD_workspace.stw**.
   - for Raisonance, select **STM8L-DISCOVERY_dev\Project\My_own_project\STVD\Raisonance\STVD_workspace.stw**.

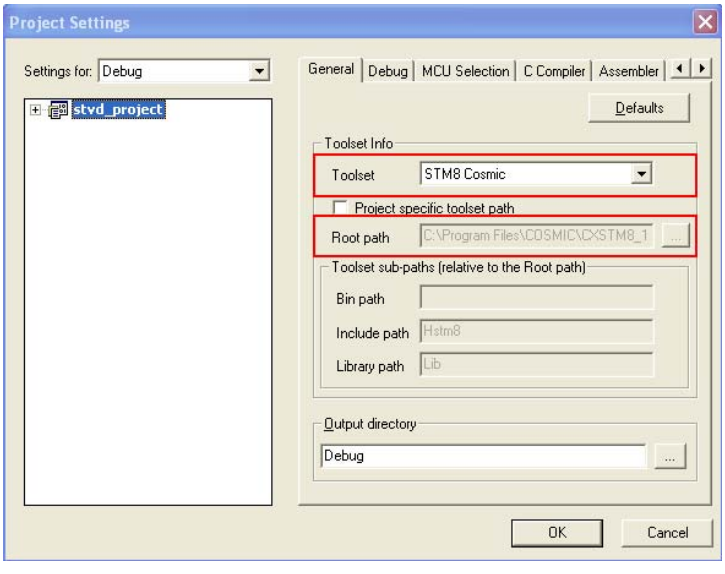
   stvd_project appears in the STVD workspace window as shown in *Figure 26*. Its structure is slightly different from the structure of **My_own_project** directory in **STM8L-Discovery_dev** package:
   - **User\src** has the same content as **STM8L-DISCOVERY_dev\Project\My_own_project\src**.
   - **STM8L15x_StdPeriph_Driver\src** contains the STM8L standard library source files.
   - **User\inc** contains **My_own_project** include files.
   - **STM8L15x_StdPeriph_Driver\src** contains the STM8L standard library include files.

Figure 26. Creating the STVD workspace for your project



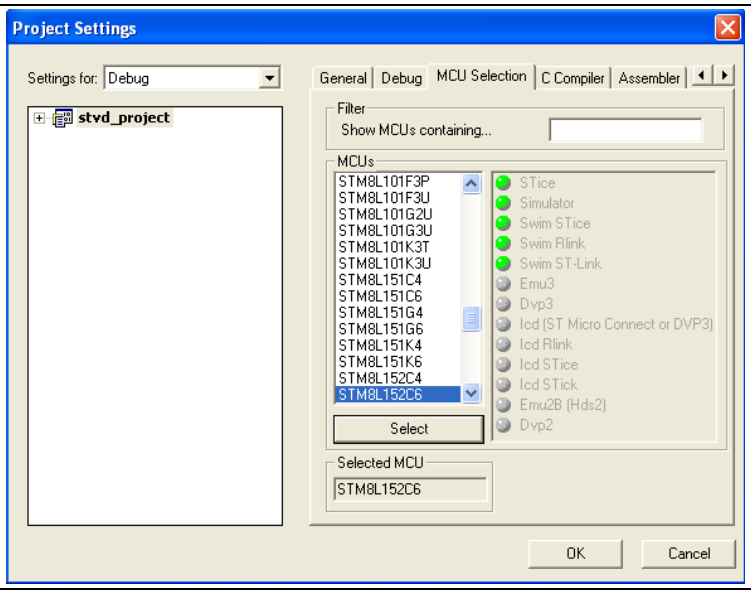
3. Check that the compiler (STM8 Cosmic or Raisonance) and its path are correct by selecting **Project>settings** from the STVD main menu (see *Figure 27*).

Figure 27. Checking the selected compiler



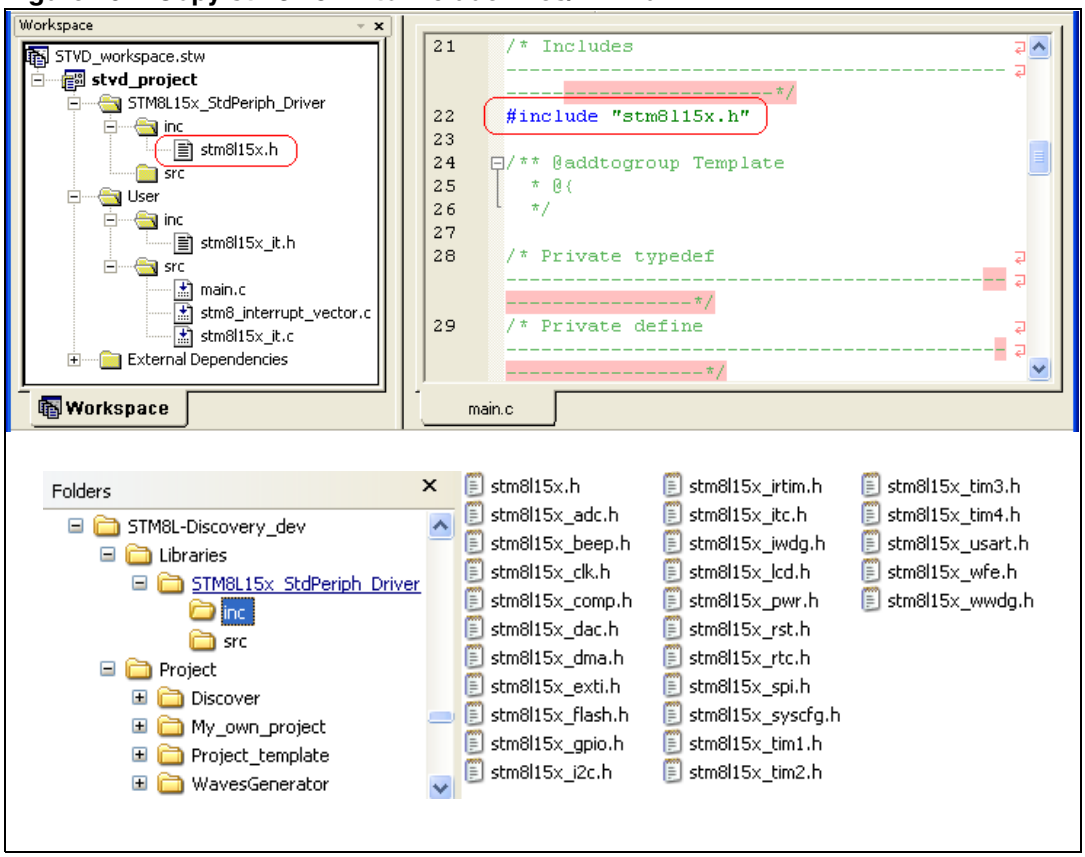
4. Go to **Projects>settings>MCU selection**, and select **STM8L152C6** from the MCUs list. Make sure that MCU selected is displayed in the selected MCU field before clicking OK (see *Figure 28*).

Figure 28. Select the MCU



5. Configure STVD to use ST-LINK as debug instrument (see *Figure 23*):
   a) Click **Debug instrument -> Target Settings**
   b) Select target **Swim ST-Link** in the target list
6. Copy *stm8l15x.h* to the **STM8L15x_StdPeriph_Driver\inc** directory of your STVD workspace (see *Figure 29*):
   – Right click on the **STM8L15x_StdPeriph_Driver\inc** directory under **stvd_project**
   – Select **Add Files to Folder**
   – Browse the *sm8l15x.h* file location in **STM8L-DISCOVERY_dev\Libraries\ STM8L15x_StdPeriph_Driver\inc**.
7. Make sure `#include <stm8l15x.h>` is present of your *main.c* file.

Figure 29. Copy *stm8l15x.h* to Include Files/FWLib



5.3.3 Linking the libraries to your STVD project

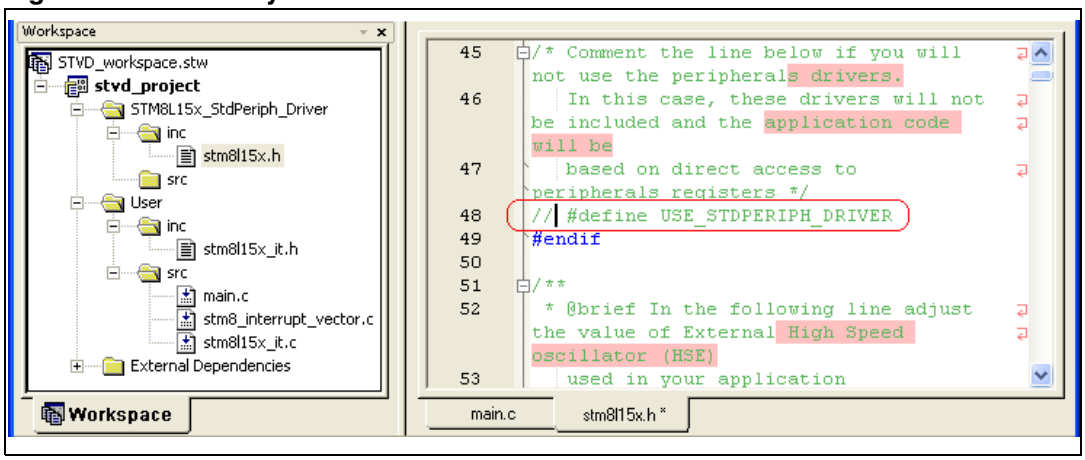
If no library linked to your STVD project

If your application does not require any library, follow the instructions below:

1. Edit *stm8l15x.h* and comment `#define USE_STDPERIPH_DRIVER` to be able to use the register structures, memory mapping, and constant definitions for each peripheral.

Note: For an example of application code that only partially uses the library, refer to the WavesGenerator project in the STM8L-DISCOVERY development package.

Figure 30. No library linked



Linking the STM8L15x standard firmware library to your STVD project

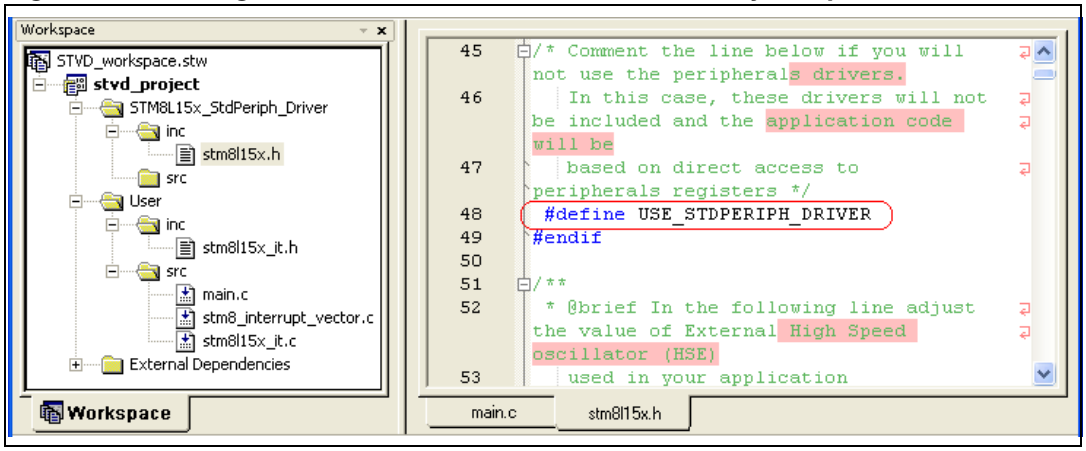
This section explains step by step how to link the STM8L standard firmware library to your STVD project.

Prior to linking the library to your project, check that the STM8L library is up-to-date. If it is not, download the latest release (see *Section 5.4: Updating the libraries*).

Follow the steps described below to link the STM8L standard firmware library to your STVD project:

1. Edit *stm8l15x.h* and uncomment `#define USE_STDPERIPH_DRIVER` (see *Figure 38*).

Figure 31. Linking the STM8L15x standard firmware library - step 1



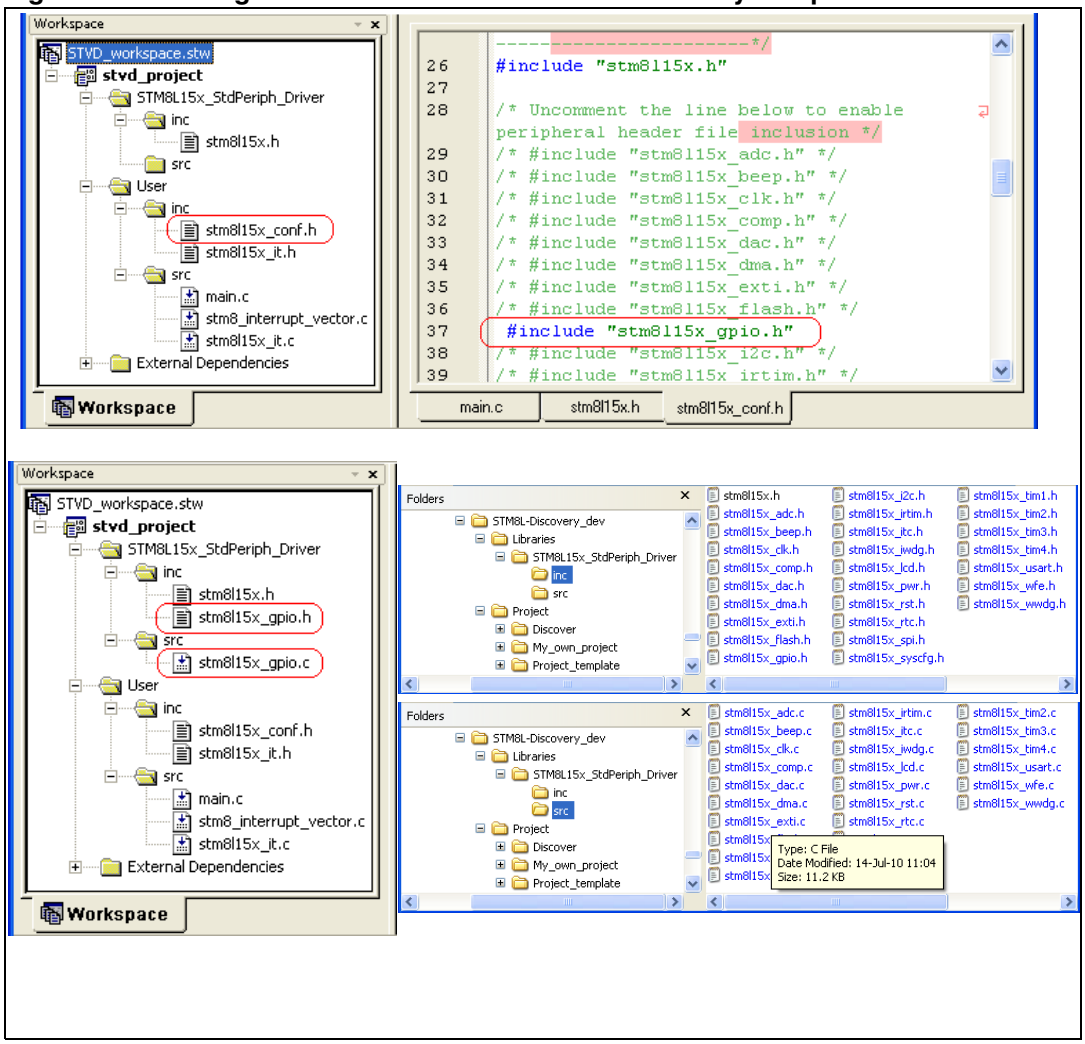
2. Copy *stm8l15x_conf.h* from the **My_own_project\inc** directory of **STM8L-DISCOVERY_dev** to **User\inc** of your STVD project.

   In *stm8l15x_conf.h*, uncomment the define statements corresponding to the peripherals used by your application.

   The .c and .h files corresponding to the peripherals which define statement are uncommented must be copied from **STM8L-DISCOVERY_dev\Libraries\STM8L15x_StdPeriph_Driver\inc** and **STM8S-DISCOVERY_dev\Libraries\STM8L15x_StdPeriph_Driver\src** into **STM8L15x_StdPeriph_Driver\inc** and **STM8L15x_StdPeriph_Driver\src** of STVD project, respectively.

   You can also decide to copy the whole drivers (.c and .h files) from the library to your STVD project.

Figure 32. Linking the STM815x standard firmware library - step 2

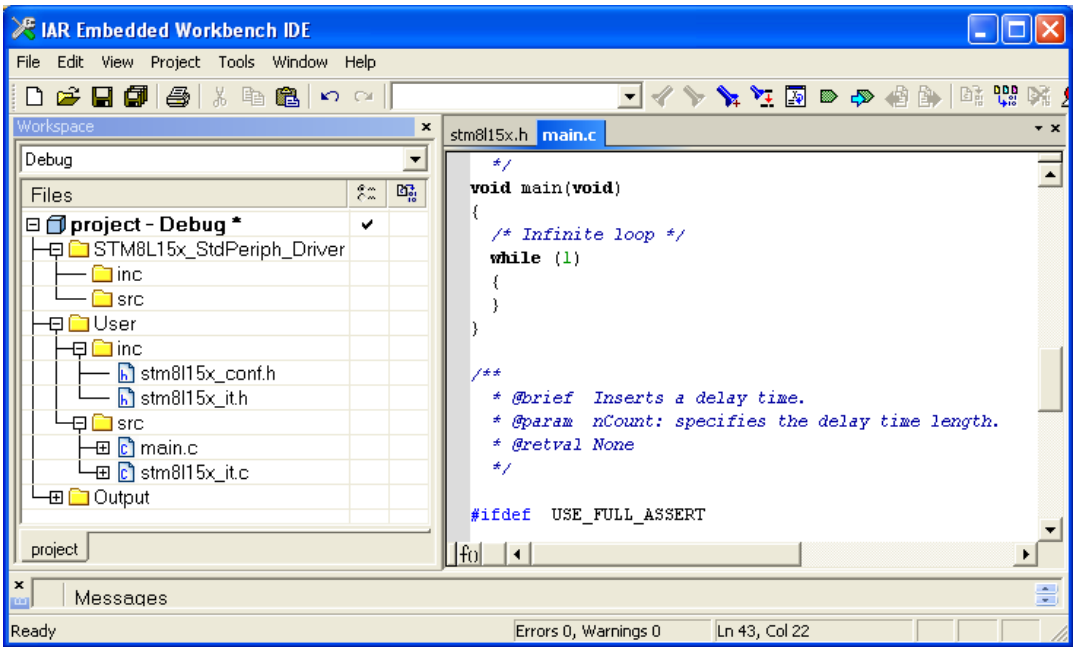


5.3.4 Creating your IAR Embedded Workbench project

1. Open IAR Embedded Workbench and click **File>Open>Workspace**
2. Browse for the **Workspace location** and select **workspace.eww** in the **EWSTM8** directory :
   - **\STM8L-DISCOVERY_dev\Project\My_own_project\EWSTM8\workspace.eww**.

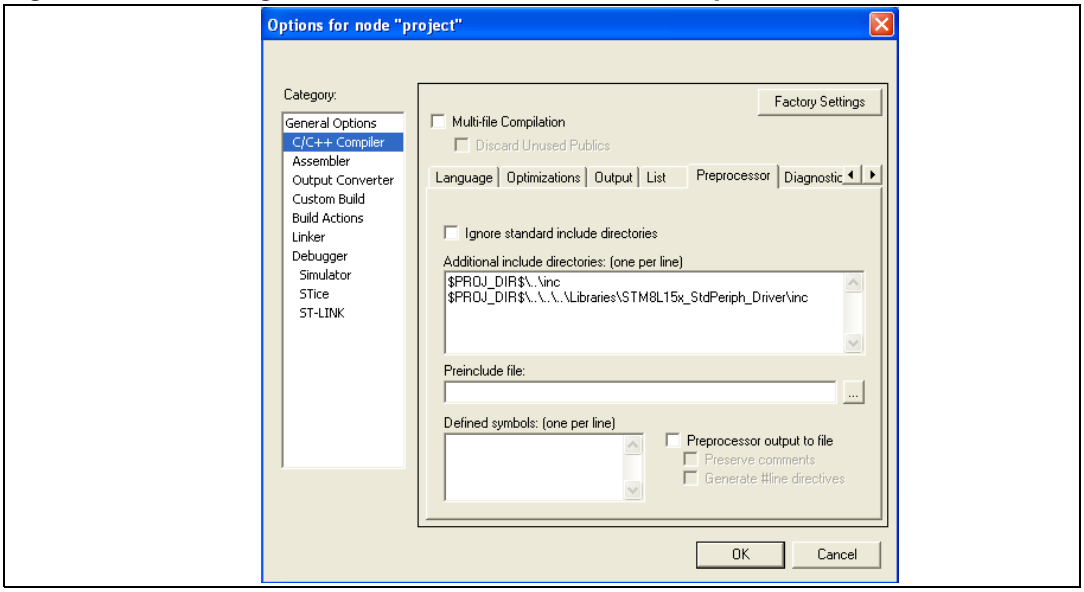
   project - Debug appears in the workspace window as shown in *Figure 26*. Its structure is slightly different from the structure of **My_own_project** directory in **STM8L-Discovery_dev** package:
   - **User\src** has the same content as **\STM8L-DISCOVERY_dev\Project\My_own_project\src.**
   - **User\inc** contains **My_own_project** include files.
   - **STM8L15x_StdPeriph_Driver\src** contains the STM8L standard library source files
   - **STM8L15x_StdPeriph_Driver\inc** contains the STM8L standard library include files

Figure 33. IAR Embedded Workbench workspace window



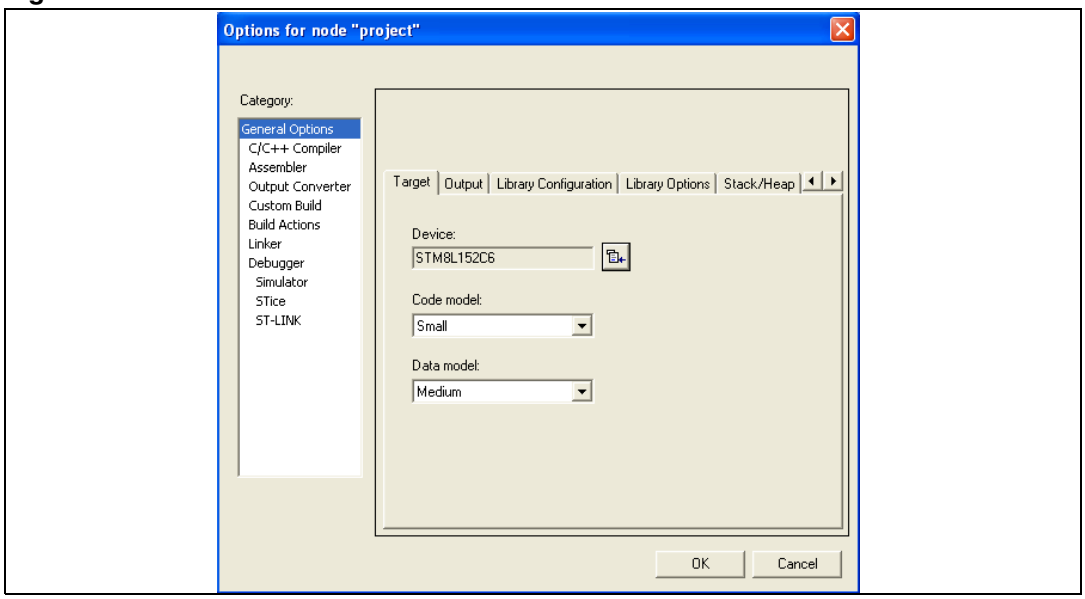
3. Check that Additional Include Directories Paths are correct. This list is to point out your project and library include files. The argument variable $PROJ_DIR$ automatically point to the directory of the current project.
   - right click on **project - Debug** in the workspace window
   - select **Options**
   - select Category C/C++ Compiler in the left window
   - select the **Preprocessor** tab
   - right andLibrary and User header files as well as their pathes are correct

Figure 34. Checking the Additional Include Directories path



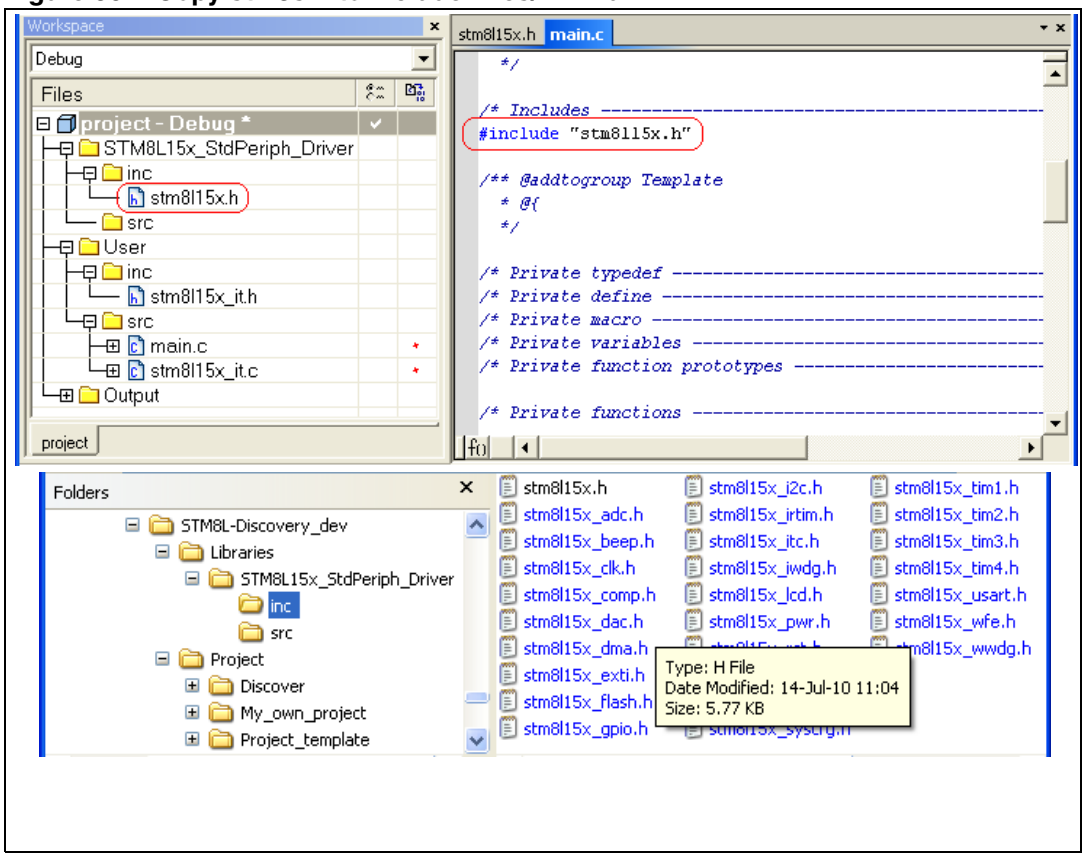
4. Select the MCU to be used: right click on **projects - Debug** in the workspace window then select **options** and **General Option** as **Category**. Then browse the Device STM8L152C6 from the STM8L list. Make sure that MCU selected is displayed in the Device field before clicking OK (see *Figure 35*).

Figure 35. Select the MCU



5. Configure IAR to use ST-LINK as debug instrument (see *Figure 23*):
   a) Right click on **project - Debug** and select **options**
   b) Select **Debugger** as Category and ST-LINK in the Driver field.
6. **Copy *stm8l15x.h* to the STM8L15x_StdPeriph_Driver/inc directory of your IAR workspace (see *Figure 29*):**
   – **Right click on the STM8L15x_StdPeriph_Driver/inc directory in the IAR workspace window.**
   – Select **Add->Add Files**
   – Browse the *sm8l15x.h* file location in **\STM8L-DISCOVERY_dev\Libraries\ STM8L15x_StdPeriph_Driver\inc**.
7. Make sure `#include <stm8l15x.h>` is present of your *main.c* file.

Figure 36. Copy *stm8s.h* to Include Files/FWLib



5.3.5 Linking the libraries to your IAR project

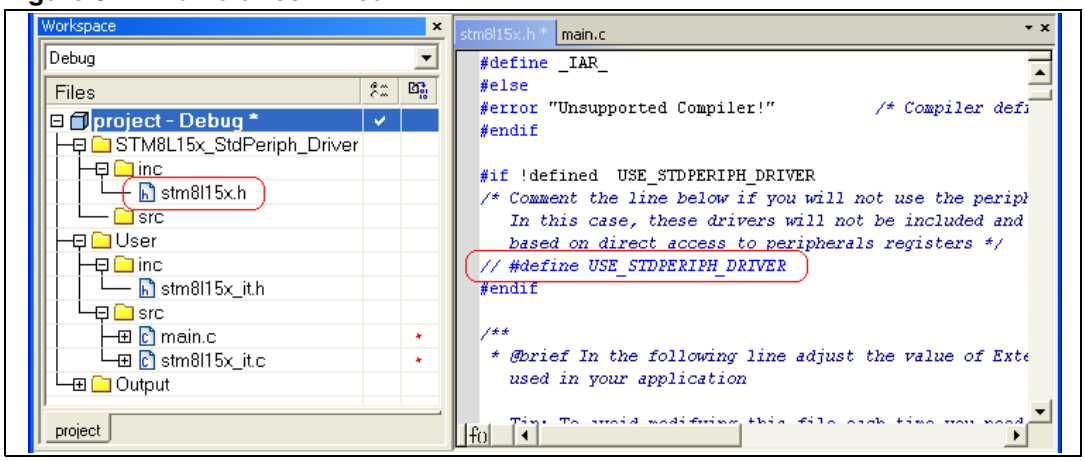
If no library linked to your IAR project

If your application does not require any library, follow the instructions below:

1. Edit *stm8l15x.h* and comment `#define USE_STDPERIPH_DRIVER` to be able to use the register structures, memory mapping, and constant definitions for each peripheral.

Note: For an example of application code that only partially uses the library, refer to the WavesGenerator project in the STM8L-DISCOVERY development package.

Figure 37. No libraries linked



Linking the STM8L standard firmware library to your IAR project

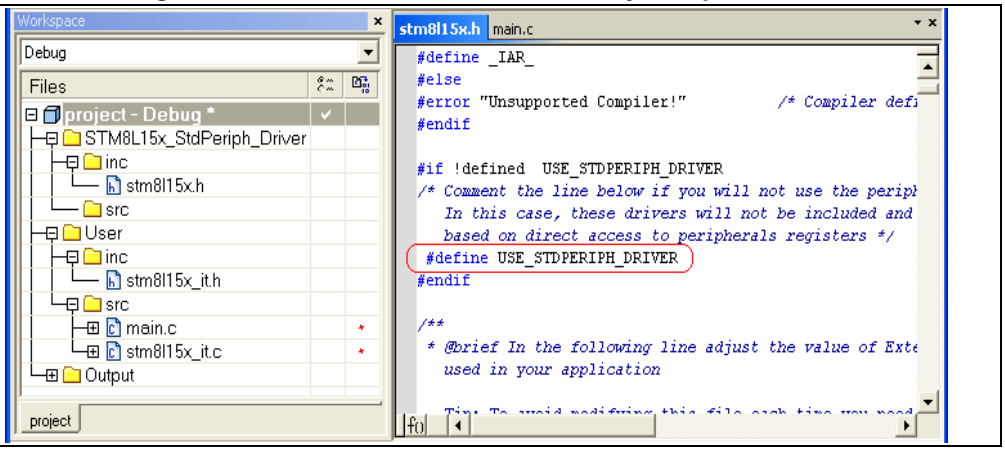
This section explains step by step how to link the STM8L standard firmware library to your IAR project.

Prior to linking the library to your project, check that the STM8L library is up-to-date. If it is not, download the latest release (see *Section 5.4: Updating the libraries*).

Follow the steps described below to link the STM8L standard firmware library to your IAR project:

1. Edit *stm8l15x.h* and uncomment `#define USE_STDPERIPH_DRIVER` (see *Figure 38*).

Figure 38. Linking the STM8L standard firmware library - step 1



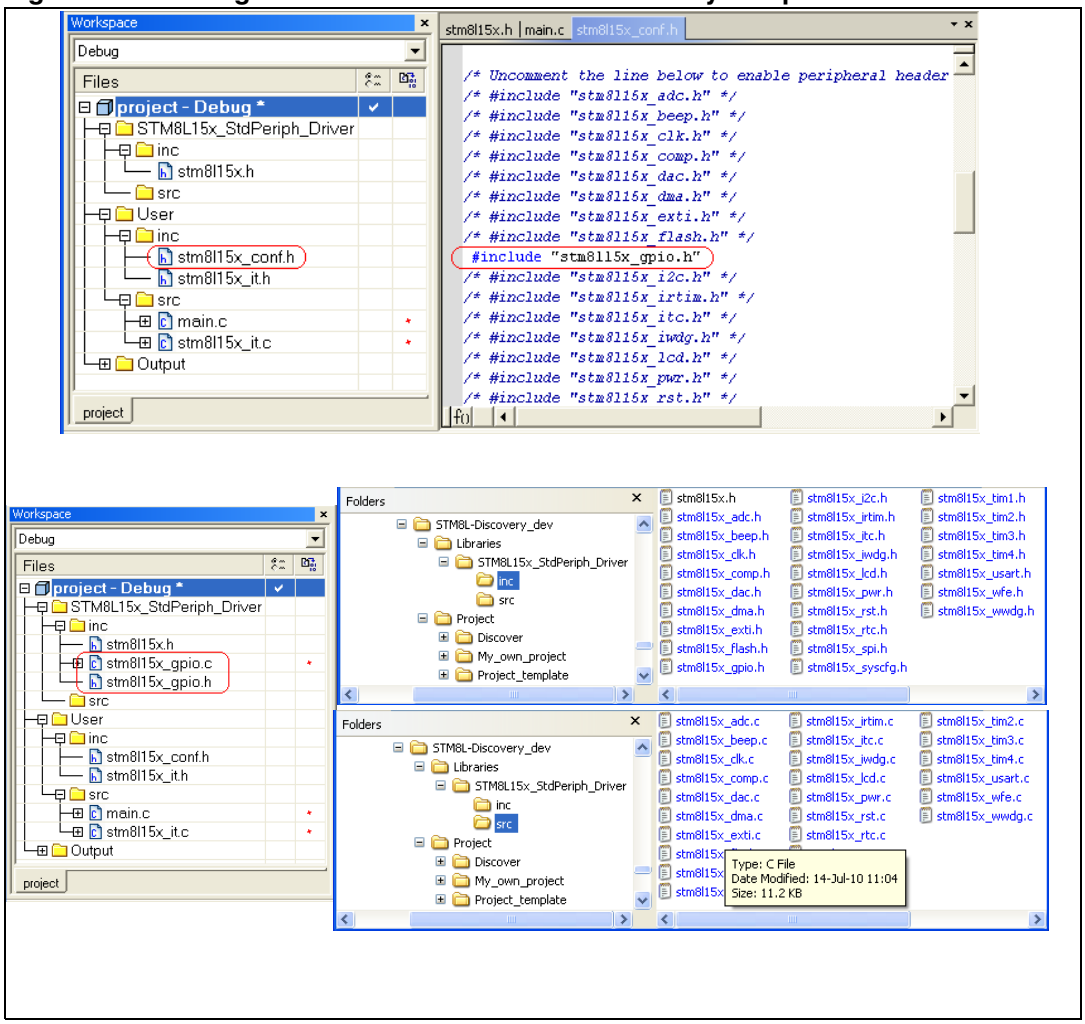
2. Copy *stm8l15x_conf.h* from the **My_own_project\inc** directory of **STM8L-DISCOVERY_dev** to **User\inc** of your IAR project.

   In *stm8l15x_conf.h*, uncomment the define statements corresponding to the peripherals used by your application.

   The .c and .h files corresponding to the peripherals for which the define statements are uncommented must be copied from **STM8L-DISCOVERY_dev\Libraries\STM8L15x_StdPeriph_Driver\inc** and **STM8L-DISCOVERY_dev\Libraries\STM8L15x_StdPeriph_Driver\src** into **STM8L15x_StdPeriph_Driver\inc** and **STM8L15x_StdPeriph_Driver\src** of IAR project, respectively.

   You can also decide to copy the whole set of drivers (.c and .h files) from the library to your STVD project.

Figure 39. Linking the STM8L standard firmware library - step 2



5.4 Updating the libraries

5.4.1 Updating the STM8L15x standard firmware library

To update the STM8L15x standard firmware library:

1. Download the new release from http://www.st.com/mcu.
2. Copy the **inc** and **src** directories in **\"STM8L-DISCOVERY_dev"\Libraries\STM8L15x_StdPeriph_Driver**.

6 Building, debugging and running your application with STVD

Once your have developed your application code, created your workspace environment, and launched STVD, you can start building, debugging and programming your application for the target microcontroller.

Note: This section refers to STVD, if you are using IAR Embedded Workbench, please refer to the IAR Getting Started documentation.

6.1 Building your application

Once your project is created, the build context is enabled by default. It allows to access all the commands required to set up, customize, and build your application.

The build configuration is available by selecting ***Build>Configurations*** from the STVP main menu toolbar. It allows you to change the application building settings.

Two preset configurations are available in STVD:

- **Debug**

  This configuration creates a version of your application that allows using all the STVD advanced debugging features. When using this configuration, output files are saved in the **Debug** directory in your **workspace** directory.

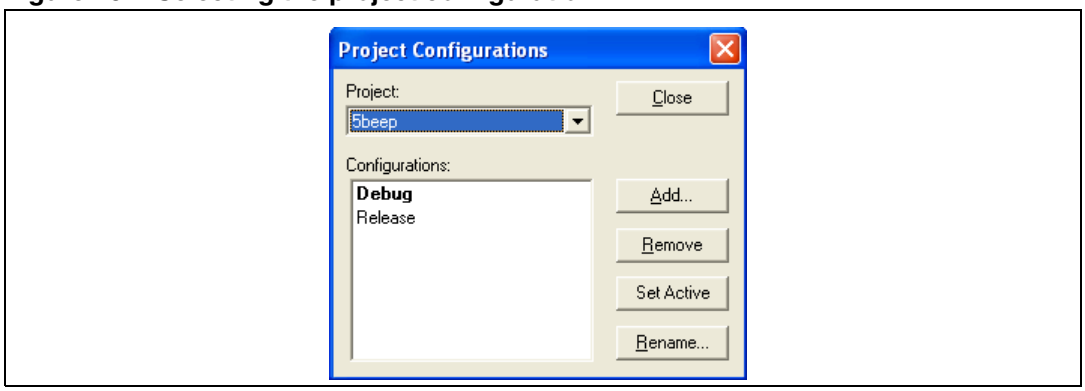
- **Release**

  This configuration creates a version of your application that uses the default optimization for your toolset. This version of your application is ready to be programmed to your target microcontroller.

Follow the sequence below to build a project:

1. Ensure that the **Debug** configuration is selected by clicking **Build>Configurations**. If the **Debug** configuration is not selected, highlight it and click **Set Active** (see *Figure 40*).

Figure 40. Selecting the project configuration



2. Once the build options are correctly configured, configure your project settings:
   a) Select ***Project > Settings*** from the STVD main menu toolbar.
   b) The **Project settings** window opens and displays all the options of your toolset compiler, assembler, and linker. You can then customize these options. For more information on the available options for your toolset, refers to the STVD online help available by clicking **Help** in the STVD main menu toolbar.
3. Once your MCU is selected (***Project > Settings > MCU Selection***) and your building options are configured, use the **Build** menu to build and rebuild your application, or compile your source files (see *Figure 41*).
4. The build command lines and possible warnings and/or errors are displayed in the **build** tab of the **Output** window (see *Figure 42*).

Figure 41. Building your project

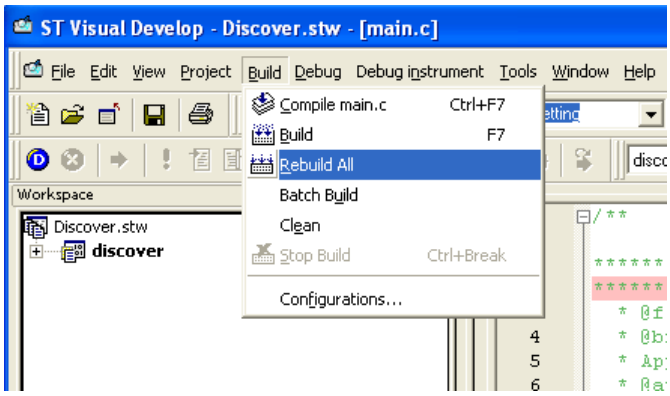
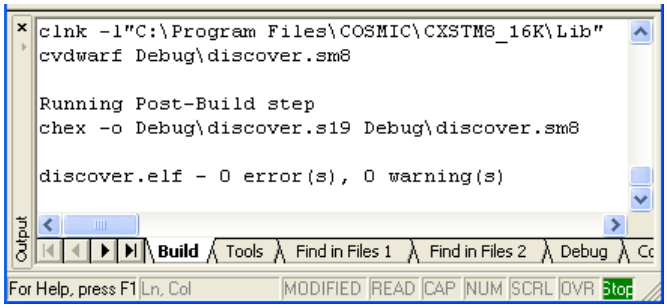


Figure 42. Building successful message

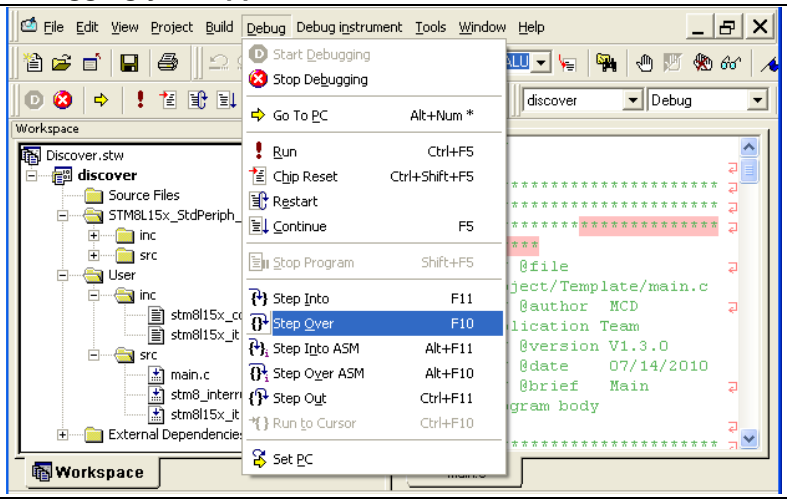

clnk -l"C:\Program Files\COSMIC\CXSTM8_16K\Lib"
cvdwarf Debug\discover.sm8
Running Post-Build step
chex -o Debug\discover.s19 Debug\discover.sm8
discover.elf - 0 error(s), 0 warning(s)
Output
Build Tools Find in Files 1 Find in Files 2 Debug
For Help, press F1 Ln, Col MODIFIED READ CAP NUM SCRL OVR

6.2 Debugging your application

Once your application has been successfully built, access the **Debug context** from the STVD main menu to access the debugging features that are supported by your debugging instrument. The debug instrument must be selected before starting debugging.

1. Select ***Debug Instrument > Target settings*** from the STVD main menu toolbar.
2. Select **Swim ST-LINK**, which is your debug instrument (see *Figure 23*).
3. Click **Apply** to confirm and **OK** to close the window.
4. Select ***Debug > Start Debugging*** to start your debug session and access the debug context. STVD then connects to your debug instrument, loads the code into the microcontroller Flash memory, and provides access to the debug commands and menus. You can now start debugging your application (see *Figure 43*).

Figure 43. Debugging your application



Several debug windows are available in STVD. You can access them by selecting **View menu** when a debug session is ongoing.

One of the most useful windows is the **Peripheral registers window.** It can be used to display the content of the STM8L105C6 peripheral registers during the debug session. (see *Figure 44*).

The **Watch window** displays the values the code variables. Just drag and drop a variable from your code into the **Watch window**.

The information displayed in the debug windows are refreshed when the program is stopped (for example by a break point). (see *Figure 45*)

Figure 44. Peripheral registers window

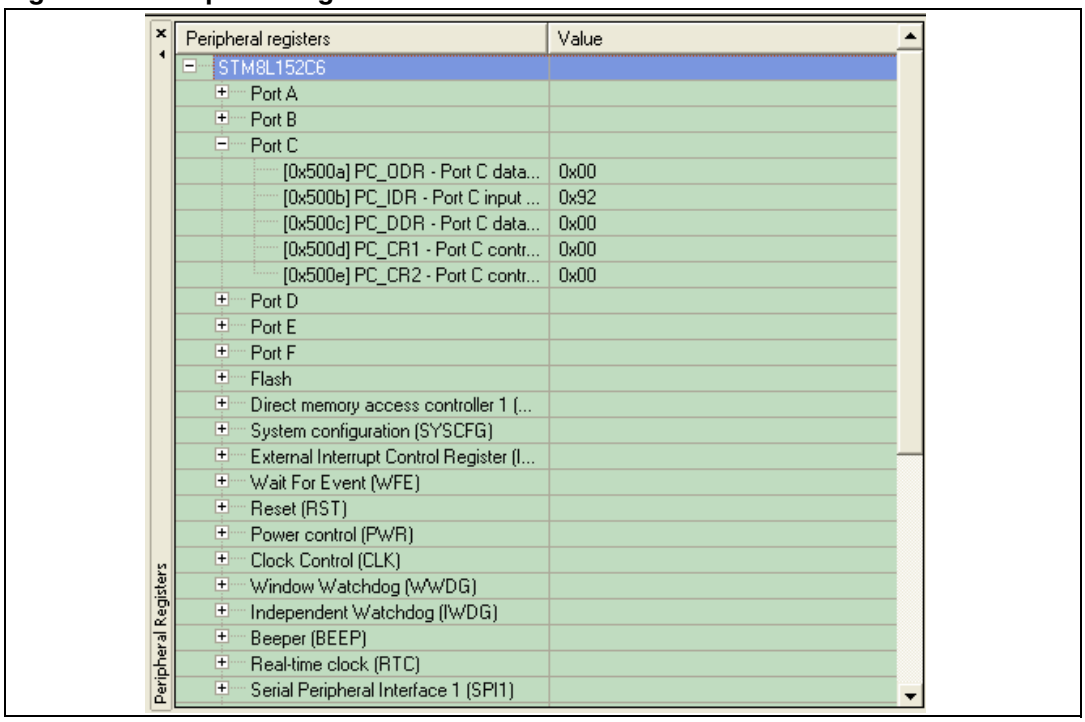
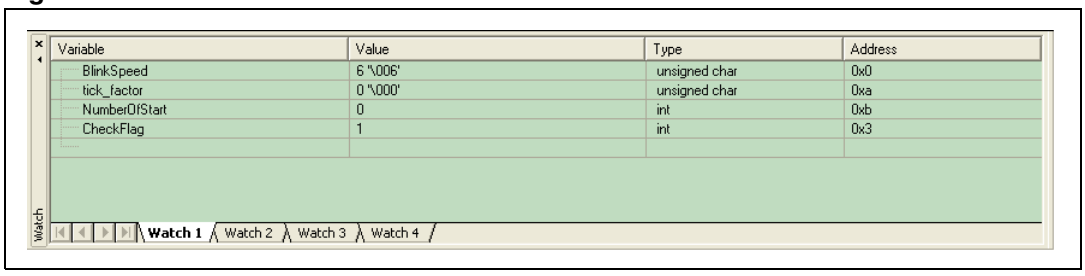


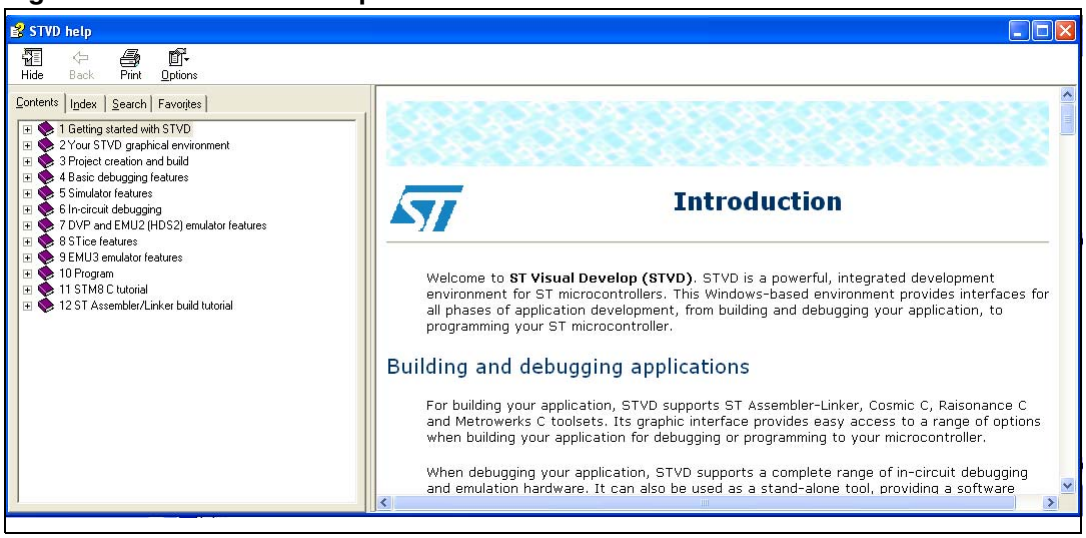
Figure 45. Watch window



6.3 Online help

For more information on building and debugging features, refer to embedded STVD online help page (**Help>Help Home Page)**.

Figure 46. STVD online help


STVD help
Hide
Back
Print
Options
Contents
Index
Search
Favorites
1 Getting started with STVD
2 Your STVD graphical environment
3 Project creation and build
4 Basic debugging features
5 Simulator features
6 In-circuit debugging
7 DVP and EMU2 (HDS2) emulator features
8 STice features
9 EMU3 emulator features
10 Program
11 STM8 C tutorial
12 ST Assembler/Linker build tutorial
Introduction
Welcome to ST Visual Develop (STVD). STVD is a powerful, integrated development environment for ST microcontrollers. This Windows-based environment provides interfaces for all phases of application development, from building and debugging your application, to programming your ST microcontroller.
Building and debugging applications
For building your application, STVD supports ST Assembler-Linker, Cosmic C, Raisonance C and Metrowerks C toolsets. Its graphic interface provides easy access to a range of options when building your application for debugging or programming to your microcontroller.
When debugging your application, STVD supports a complete range of in-circuit debugging and emulation hardware. It can also be used as a stand-alone tool, providing a software

7 Revision history

Table 1. Document revision history

Date	Revision	Changes
14-Sep-2010	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2010 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com